

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2000134230 A**

(43) Date of publication of application: 12 . 05 . 00

(51) Int. Cl.

H04L 12/40

(21) Application number: 10321489

(22) Date of filing: 27 . 10 . 98

(71) Applicant: **SEIKO EPSON CORP**

(72) Inventor: **KANBARA YOSHIYUKI**
ISHIDA TAKUYA
WADA FUMITOSHI

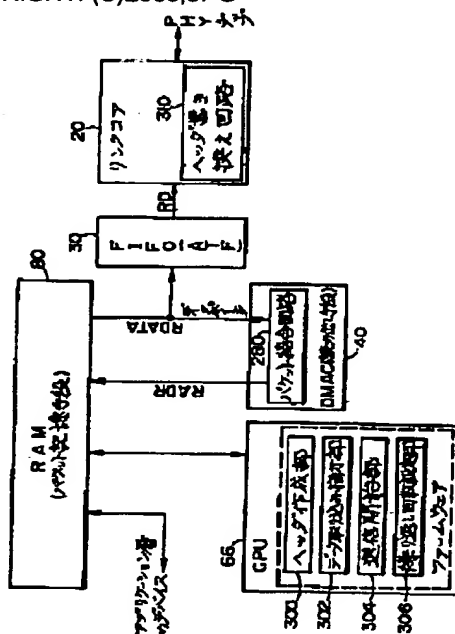
(54) DATA TRANSFER CONTROLLER AND ELECTRONIC EQUIPMENT

COPYRIGHT: (C)2000,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a data transfer controller and electronic equipment capable of reducing the overheads of processings and transferring data at high speed with the hardware of a small scale.

SOLUTION: Relating to the data transfer of IEEE1394, a packet connection circuit 280 reads the header and data of packets from the header and data areas of a RAM 80 and connects them. By utilizing the generation period of header CRC, a data pointer is obtained. Which one of the header and the data is read is judged by a code and a header pointer or the data pointer is increased. The header is generated while fetching the data to the data area. While the packets are transmitted from one channel for separating a transmission packet area, the data are fetched to the other channel. By using a connection pointer, the packets of the other channel are successively read. An ACK code from a transfer destination is written back to the channel of a transmission origin. A basic header is rewritten, the header is successively prepared and the packets are successively transferred until the number of times of repetition becomes zero.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-134230

(P2000-134230A)

(43) 公開日 平成12年5月12日 (2000.5.12)

(51) Int.Cl.⁷

H 0 4 L 12/40

識別記号

F I

H 0 4 L 11/00

テーマコード(参考)

3 2 0

5 K 0 3 2

審査請求 未請求 請求項の数16 F D (全 30 頁)

(21) 出願番号

特願平10-321489

(22) 出願日

平成10年10月27日 (1998. 10. 27)

(71) 出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72) 発明者 神原 義幸

長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

(72) 発明者 石田 卓也

長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

(74) 代理人 100090479

弁理士 井上 一 (外2名)

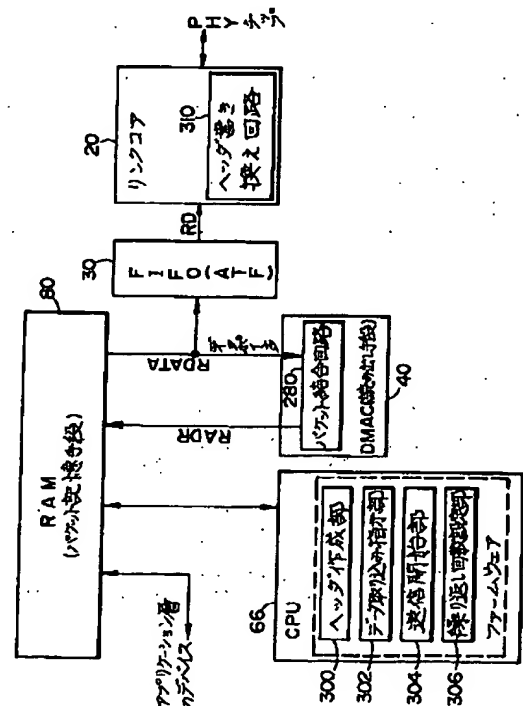
最終頁に続く

(54) 【発明の名称】 データ転送制御装置及び電子機器

(57) 【要約】

【課題】 処理のオーバーヘッドを軽減し、小規模なハードウェアで高速なデータ転送を実現できるデータ転送制御装置及び電子機器を提供すること。

【解決手段】 IEEE1394のデータ転送において、パケット結合回路280が、パケットのヘッダ、データをRAM80のヘッダ、データ領域から読み出し結合する。ヘッダCRCの生成期間を利用してデータポイントを取得する。ヘッダ、データのいずれが読み出されているかをt c o d eにより判断し、ヘッダポイント又はデータポイントをインクリメントする。データ領域へのデータ取り込みの間にヘッダを生成する。送信パケット領域を分離する1のチャンネルからパケットが送信されている間に他のチャンネルへデータを取り込む。連結ポイントを用いて他のチャンネルのパケットを連続して読み出す。転送先からのACKコードを送信元のチャンネルに書き戻す。基本ヘッダを書き換えてヘッダを順次作成し、繰り返し回数が零になるまでパケットを連続転送する。



【特許請求の範囲】

【請求項 1】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、ランダムアクセス可能な記憶手段が、上層により制御情報が書き込まれている制御情報領域と上層によりデータが書き込まれているデータ領域とに分離されている場合において、パケットの制御情報を前記記憶手段の前記制御情報領域から読み出し、該制御情報と対をなすパケットのデータを前記記憶手段の前記データ領域から読み出すパケット結合手段と、読み出されたパケットを各ノードに転送するためのサービスを提供するリンク手段とを含むことを特徴とするデータ転送制御装置。

【請求項 2】 請求項 1 において、前記パケット結合手段が、前記データ領域から読み出されるデータのアドレスを指すデータポインタを、前記制御情報領域から読み出された制御情報から取得し、取得したデータポインタを用いて前記データ領域からデータを読み出すことを特徴とするデータ転送制御装置。

【請求項 3】 請求項 2 において、前記パケット結合手段が、パケットの制御情報のエラーチェック情報を前記リンク手段が生成する期間を利用して、制御情報からデータポインタを取得することを特徴とするデータ転送制御装置。

【請求項 4】 請求項 1 乃至 3 のいずれかにおいて、前記パケット結合手段が、パケットの制御情報が読み出されていると、パケットの制御情報に含まれるパケットフォーマット識別情報に基づき判断した場合には、前記制御情報領域から読み出される制御情報のアドレスを指す制御情報ポインタを更新し、パケットのデータが読み出されていると、前記パケットフォーマット識別情報に基づき判断した場合には、前記データ領域から読み出されるデータのアドレスを指すデータポインタを更新することを特徴とするデータ転送制御装置。

【請求項 5】 請求項 1 乃至 4 のいずれかにおいて、前記データ領域へのデータの取り込み処理の間に、制御情報を作成し該制御情報を前記制御情報領域に書き込む制御情報作成手段と、データの取り込み処理と制御情報の書き込み処理の両方が完了したことを条件に、パケットの送信開始を命令する送信開始手段とを含むことを特徴とするデータ転送制御装置。

【請求項 6】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、ランダムアクセス可能な記憶手段の送信パケット領域が複数のチャンネルに分離されている場合において、前記複数のチャンネルに格納されるパケットの中で、データの取

り込み処理と制御情報の書き込み処理の両方が完了したパケットの送信開始を命令する送信開始手段と、送信開始が命令されたパケットを、送信パケット領域の対応するチャンネルから読み出す読み出し手段と、読み出されたパケットを各ノードに転送するためのサービスを提供するリンク手段とを含むことを特徴とするデータ転送制御装置。

【請求項 7】 請求項 6 において、前記複数のチャンネルの中の 1 のチャンネルからパケットが送信されている間に、他のチャンネルのパケットについてのデータの取り込みを指示する手段を含むことを特徴とするデータ転送制御装置。

【請求項 8】 請求項 6 又は 7 において、データ取り込み中フラグがクリアされていることを条件に、データの取り込みを指示すると共に該データ取り込み中フラグをセットし、データの取り込みが終了したことを条件に、該データ取り込み中フラグをクリアする手段と、

送信中フラグがクリアされていることを条件に、パケットの送信開始を命令すると共に該送信中フラグをセットし、パケットの送信が終了したことを条件に、該送信中フラグをクリアする手段とを含むことを特徴とするデータ転送制御装置。

【請求項 9】 請求項 6 乃至 8 のいずれかにおいて、各チャンネルに格納される各パケットが、パケット同士を連結するための連結ポインタを含み、前記読み出し手段が、

前記送信開始手段により 1 のチャンネルのパケットの送信開始が命令された場合に、前記連結ポインタを用いて、他のチャンネルのパケットを連続して順次読み出すことを特徴とするデータ転送制御装置。

【請求項 10】 請求項 6 乃至 9 のいずれかにおいて、パケットの転送先から返送されてくるアクノリッジメント情報を、送信パケット領域の複数のチャンネルの中の該パケットの送信元のチャンネルに書き戻す書き戻し手段を含むことを特徴とするデータ転送制御装置。

【請求項 11】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、記憶手段に書き込まれたパケットを読み出す読み出し手段と、

読み出されたパケットを各ノードに転送するためのサービスを提供するリンク手段と、パケットの転送先から返送されてくるアクノリッジメント情報を、返送されてきたアクノリッジメント情報を確認せずに連続転送できるパケット数以上の個数だけ記憶する手段とを含むことを特徴とするデータ転送制御装置。

【請求項 12】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、送信繰り返し回数情報を設定する手段と、

基本制御情報を作成する制御情報作成手段と、
パケットの送信開始を命令する送信開始手段と、
パケットの送信開始が命令された場合に、前記基本制御
情報を書き換えることで、連続転送されるパケットのデ
ータと対をなす制御情報を順次生成する制御情報書き換
え手段と、
データと順次生成された制御情報とによりフレームが構
成されるパケットを、送信繰り返し回数情報が所与の値
になるまで連続転送する手段とを含むことを特徴とするデ
ータ転送制御装置。

【請求項13】 請求項12において、
前記基本制御情報が、
前記送信繰り返し回数情報と、ランダムアクセス可能な
記憶手段のデータ領域から読み出されるデータのアドレ
スを指すデータポインタと、トランザクション識別情報
とを含み、
前記制御情報書き換え手段が、
前記送信繰り返し回数情報、前記データポインタ、及び
前記トランザクション識別情報を書き換えることを特徴
とするデータ転送制御装置。

【請求項14】 請求項1乃至13のいずれかにおい
て、
IEEE1394の規格に準拠したデータ転送を行うこ
とを特徴とするデータ転送制御装置。

【請求項15】 請求項1乃至14のいずれかのデータ
転送制御装置と、
前記データ転送制御装置及びバスを介して他のノードか
ら受信したデータに所与の処理を施す装置と、
処理が施されたデータを出力又は記憶するための装置と
を含むことを特徴とする電子機器。

【請求項16】 請求項1乃至14のいずれかのデータ
転送制御装置と、
前記データ転送制御装置及びバスを介して他のノードに
送信するデータに所与の処理を施す装置と、
処理が施されるデータを取り込むための装置とを含むこ
とを特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ転送制御装
置及びこれを含む電子機器に関する。

【0002】

【背景技術及び発明が解決しようとする課題】近年、I
EEE1394と呼ばれるインターフェース規格が脚光
を浴びている。このIEEE1394は、次世代のマル
チメディアにも対応可能な高速シリアルバスインターフ
ェースを規格化したものである。このIEEE1394
によれば、動画像などのリアルタイム性が要求されるデ
ータも扱うことができる。また、IEEE1394のバ
スには、プリンタ、スキャナ、CD-Rドライブ、ハー
ドディスクドライブなどのコンピュータの周辺機器のみ

ならず、ビデオカメラ、VTR、TVなどの家庭用電化
製品も接続できる。このため、電子機器のデジタル化を
飛躍的に促進できるものとして期待されている。

【0003】このようなIEEE1394の概要につい
ては、例えば「IEEE1394ハイ・パフォーマンス
・シリアルバスの概要」(Interface Ap
r. 1996の1~10頁)、「PC周辺機器用バス規
格群総覧」(Interface Jan. 1997の
106頁~116頁)、「IEEE1394-1995
(FireWire)のリアルタイム転送モードとマル
チメディア対応プロトコル」(Interface J
an. 1997の136~146頁)に開示されてい
る。また、IEEE1394に準拠したデータ転送制御
装置としては、テキサス・インスツルメンツ社製のTS
B12LV31などが知られている。

【0004】しかしながら、このIEEE1394に準
拠したデータ転送制御装置には次のような課題があるこ
とが判明した。

【0005】即ち、現在のIEEE1394規格によれ
ば最大で400Mbpsの転送速度が実現可能となっ
ている。しかし、現実には、処理のオーバーヘッドの存在
に起因して、システム全体の実転送速度はこれよりもか
なり低くなっている。つまり、CPU上で動作するファ
ームウェアやアプリケーションソフトが、送信データを
準備したり、受信データを取り込んだりするなどの処理
に多くの時間を要してしまい、バス上での転送速度が速
くても、結局、高速なデータ転送を実現できない。

【0006】特に、周辺機器に組み込まれるCPUは、
パーソナルコンピュータなどのホストシステムに組み込
まれるCPUに比べて処理能力が低い。このため、ファ
ームウェアやアプリケーションソフトの処理のオーバー
ヘッドの問題は、非常に深刻なものとなる。従って、こ
のようなオーバーヘッドの問題を効果的に解消できる技
術が望まれている。

【0007】本発明は、以上のような技術的課題に鑑み
てなされたものであり、その目的とするところは、ファ
ームウェアやアプリケーションソフトなどの処理のオー
バーヘッドを軽減し、小規模なハードウェアで高速なデ
ータ転送を実現できるデータ転送制御装置及びこれが用
いられる電子機器を提供することにある。

【0008】

【課題を解決するための手段】上記課題を解決するた
めに本発明は、バスに接続される複数のノード間でのデ
ータ転送のためのデータ転送制御装置であって、ランダム
アクセス可能な記憶手段が、上層によりパケットの制御
情報が書き込まれている制御情報領域と上層によりパケ
ットのデータが書き込まれているデータ領域とに分離さ
れている場合において、パケットの制御情報を前記記憶
手段の前記制御情報領域から読み出し、該制御情報と対
をなすパケットのデータを前記記憶手段の前記データ領

域から読み出すパケット結合手段と、読み出されたパケットを各ノードに転送するためのサービスを提供するリンク手段とを含むことを特徴とする。

【0009】本発明によれば、ランダムアクセス可能な記憶手段が制御情報領域とデータ領域とに分離されている。そして、制御情報領域には、例えばファームウェアなどの上層により制御情報が書き込まれ、データ領域には、例えばアプリケーションソフトやファームウェアなどの上層によりデータが書き込まれている（取り込まれている）。そして、本発明では、パケット結合手段が、制御情報領域から制御情報を読み出すと共に、データ領域から、この制御情報と対をなすデータを読み出し結合する。そして、制御情報とデータとによりフレームが構成されるパケットが、リンク手段を介して各ノードに転送される。

【0010】本発明によれば、上層は、パケットを送信する順序に依存せずに、制御情報やデータを記憶手段に書き込める。また、上層は、送信すべきパケットのデータをデータ領域に連続して書き込むこともできる。また、制御情報領域の制御情報とデータ領域のデータとを結合してパケットを組み立てる処理に、上層が関与する必要がない。以上により、本発明によれば、上層の処理負担を格段に軽減できるようになる。

【0011】また本発明は、前記パケット結合手段が、前記データ領域から読み出されるデータのアドレスを指すデータポインタを、前記制御情報領域から読み出された制御情報から取得し、取得したデータポインタを用いて前記データ領域からデータを読み出すことを特徴とする。このようにすれば、制御情報と対になるデータを適正にデータ領域から読み出せるようになると共に、制御情報とデータとを結合しパケットを組み立てる処理の簡素化を図れる。

【0012】また本発明は、前記パケット結合手段が、パケットの制御情報のエラーチェック情報を前記リンク手段が生成する期間を利用して、制御情報からデータポインタを取得することを特徴とする。このようにすれば、処理時間に無駄が生じるのを防止でき、処理の高速化を図れる。

【0013】また本発明は、前記パケット結合手段が、パケットの制御情報が読み出されていると、パケットの制御情報に含まれるパケットフォーマット識別情報に基づき判断した場合には、前記制御情報領域から読み出される制御情報のアドレスを指す制御情報ポインタを更新し、パケットのデータが読み出されていると、前記パケットフォーマット識別情報に基づき判断した場合には、前記データ領域から読み出されるデータのアドレスを指すデータポインタを更新することを特徴とする。このようにすれば、制御情報ポインタの更新処理とデータポインタの更新処理を、パケットフォーマット識別情報を利用して切り替え、パケットの制御情報とデータとを結合

できるようになる。これにより、制御情報とデータの結合処理の簡素化を図れる。

【0014】また本発明は、前記データ領域へのデータの取り込み処理の間に、制御情報を作成し該制御情報を前記制御情報領域に書き込む制御情報作成手段と、データの取り込み処理と制御情報の書き込み処理の両方が完了したことを条件に、パケットの送信開始を命令する送信開始手段とを含むことを特徴とする。このようにすればデータの取り込み処理と、制御情報の生成及び書き込み処理とを、並列に行うことができ、処理の効率化を図れるようになる。

【0015】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、ランダムアクセス可能な記憶手段の送信パケット領域が複数のチャンネルに分離されている場合において、前記複数のチャンネルに格納されるパケットの中で、データの取り込み処理と制御情報の書き込み処理の両方が完了したパケットの送信開始を命令する送信開始手段と、送信開始が命令されたパケットを、送信パケット領域の対応するチャンネルから読み出す読み出し手段と、読み出されたパケットを各ノードに転送するためのサービスを提供するリンク手段とを含むことを特徴とする。

【0016】本発明によれば、記憶手段の送信パケット領域が複数のチャンネルに分離される。そして、パケットの準備（データの取り込み処理及び制御情報の生成及び書き込み処理）が完了し、送信開始手段が送信開始を命令すると、対応するチャンネルからパケットが読み出され、送信される。従って、本発明によれば、1のチャンネルからのパケットの送信開始を命令した後に、その送信の完了を待つことなく、他のチャンネルのパケットを準備することなどが可能になる。この結果、処理時間の無駄を無くすことができ、処理の効率化を図れる。

【0017】また本発明は、前記複数のチャンネルの中の1のチャンネルからパケットが送信されている間に、他のチャンネルのパケットについてのデータの取り込みを指示する手段を含むことを特徴とする。このようにすれば、1のチャンネルでのパケットの送信処理と、他のチャンネルでのデータの取り込み処理とを並列に行うことが可能になる。この結果、複数のパケットを連続して送信する場合に、処理を大幅に効率化できる。

【0018】また本発明は、データ取り込み中フラグがクリアされていることを条件に、データの取り込みを指示すると共に該データ取り込み中フラグをセットし、データの取り込みが終了したことを条件に、該データ取り込み中フラグをクリアする手段と、送信中フラグがクリアされていることを条件に、パケットの送信開始を命令すると共に該送信中フラグをセットし、パケットの送信が終了したことを条件に、該送信中フラグをクリアする手段とを含むことを特徴とする。このようにデータ取り込み中フラグや送信中フラグを用いることで、1のチャ

ネルのデータ取り込み処理が行われている間に他のチャネルのデータ取り込み処理が行われたり、1のチャネルの送信処理が行われている間に他のチャネルの送信処理が行われてしまうという事態を回避できる。そして、1つのプログラムを用いて、複数チャネルからのパケットの送信処理をマルチタスクのように実行することが可能になる。

【0019】また本発明は、各チャネルに格納される各パケットが、パケット同士を連結するための連結ポイントを含み、前記読み出し手段が、前記送信開始手段により1のチャネルのパケットの送信開始が命令された場合に、前記連結ポイントを用いて、他のチャネルのパケットを連続して順次読み出すことを特徴とする。このようにすれば、1のチャネルのパケットの送信開始を命令するだけで、他のチャネルのパケットが連続して順次読み出され送信されるようになり、ファームウェアなどの上層の処理負担を格段に軽減できる。

【0020】また本発明は、パケットの転送先から返送されてくるアクリッジメント情報を、送信パケット領域の複数のチャネルの中の該パケットの送信元のチャネルに書き戻す書き戻し手段を含むことを特徴とする。このようにすれば、送信パケットと、その転送先から返ってくるアクリッジメント情報とを一対一に対応させることが可能になる。これにより、どのアクリッジメント情報がどのパケットに対応するのかを、ファームウェアなどの上層が容易に認識できるようになる。

【0021】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、記憶手段に書き込まれたパケットを読み出す読み出し手段と、読み出されたパケットを各ノードに転送するためのサービスを提供するリンク手段と、パケットの転送先から返送されてくるアクリッジメント情報を、返送されてきたアクリッジメント情報を確認せずに連続転送できるパケット数以上の個数だけ記憶する手段とを含むことを特徴とする。このようにすれば、ファームウェアなどの上層は、返送されてくるアクリッジメント情報を確認することなく次々にパケットの送信開始を命令できるようになる。この結果、処理の効率化を図れる。

【0022】また本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、送信繰り返し回数情報を設定する手段と、基本制御情報を作成する制御情報作成手段と、パケットの送信開始を命令する送信開始手段と、パケットの送信開始が命令された場合に、前記基本制御情報を書き換えることで、連続転送されるパケットのデータと対をなす制御情報を順次生成する制御情報書き換え手段と、データと順次生成された制御情報とによりフレームが構成されるパケットを、送信繰り返し回数情報が所与の値になるまで連続転送する手段とを含むことを特徴とする。

【0023】本発明によれば、送信繰り返し回数情報を設定すると共に基本制御情報を作成し、送信開始を命令すると、自動的に基本制御情報から制御情報が生成され、送信繰り返し回数情報が所与の値（例えば零）になるまでパケットが連続転送される。これにより、ファームウェアの処理負担を増すことなく、複数のパケットを連続転送し、大きなサイズのデータを他のノードに転送できるようになる。

【0024】なお本発明では、前記基本制御情報が、前記送信繰り返し回数情報と、ランダムアクセス可能な記憶手段のデータ領域から読み出されるデータのアドレスを指すデータポイントと、トランザクション識別情報とを含み、前記制御情報書き換え手段が、前記送信繰り返し回数情報、前記データポイント、及び前記トランザクション識別情報を書き換えることが望ましい。

【0025】また本発明では、IEEE1394の規格に準拠したデータ転送を行うことが望ましい。

【0026】また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする。また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、処理が施されるデータを取り込むための装置とを含むことを特徴とする。

【0027】本発明によれば、他のノードから転送されたデータを電子機器において出力したり記憶したりする処理、電子機器において取り込んだデータを他のノードに転送したりする処理を高速化することが可能になる。また、本発明によれば、データ転送制御装置を小規模化できると共に、データ転送を制御するファームウェアなどの処理負担を軽減できるため、電子機器の低コスト化、小規模化などを図ることも可能になる。

【0028】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0029】1. IEEE1394

まず、IEEE1394の概要について簡単に説明する。

【0030】1. 1 データ転送速度、接続トポロジー
IEEE1394 (IEEE1394-1995、IEEE1394.A) では100~400Mbpsの高速なデータ転送が可能となっている (IEEE1394.Bでは800~3200Mbps)。また、転送速度が異なるノードをバスに接続することも許される。

【0031】各ノードはツリー状に接続されており、1つのバスに最大で63個のノードが接続可能になっている。なお、バスブリッジを利用すれば約64000個の

ノードを接続することも可能である。

【0032】電源が投入されたり、途中でデバイスの抜き差しが発生すると、バスリセットが発生し、接続トポロジに関する情報が全てクリアされる。そして、バスリセット後、ツリー識別（ルートノードの決定）、自己識別が行われる。その後、アイソクロナスリソースマネージャ、サイクルマスタ、バスマネージャ等の管理ノードが決定される。そして、通常の packets 転送が開始される。

【0033】1. 2 転送方式

IEEE 1394 では、packets の転送方式として、信頼性が要求されるデータの転送に好適な非同期転送と、リアルタイム性が要求される動画や音声などのデータの転送に好適なアイソクロナス転送が用意されている。

【0034】図 1 (A) に、非同期サブアクションの例を示す。1つのサブアクションは、調停、packets 転送、acknowledgment からなる。即ち、データ転送に先立って、まず、バスの使用権に関する調停が行われる。そしてソース（転送元）ノードから destination（転送先）ノードに packets が転送される。この packets のヘッダにはソース ID と destination ID が含まれる。destination ノードは、この destination ID を読んで、自ノード宛の packets か否かを判断する。destination ノードは、packets を受け取ると、ソースノードに acknowledgment（ACK）の packets を返す。

【0035】packets 転送と ACK の間には acknowledgment gap が存在する。また、1つのサブアクションと次のサブアクションの間にはサブアクション gap が存在する。そして、サブアクション gap に相当する一定のバス・アイドル時間が経過しないと、次のサブアクションの調停を開始できない。これによりサブアクション相互の衝突が回避される。

【0036】図 1 (B) に、アイソクロナスサブアクションの例を示す。アイソクロナス転送はブロードキャスト（バスに接続される全てのノードに転送）で実行されるため、packets 受信時に ACK は返送されない。また、アイソクロナス転送では、ノード ID ではなくチャンネル番号を使用して packets 転送が行われる。なお、サブアクション間にはアイソクロナス gap が存在する。

【0037】図 1 (C) に、データ転送時のバスの様子を示す。アイソクロナス転送は、サイクルマスタが一定周期毎にサイクルスタート packets を発生することで開始する。これにより、1つのチャンネル当たり、125 μ s 毎に少なくとも 1つの packets を転送できるようになる。この結果、動画や音声などのリアルタイム性が要求されるデータの転送が可能になる。

【0038】非同期転送はアイソクロナス転送の合間に行われる。即ち、アイソクロナス転送の方が非同期転送

よりも優先順位が高くなっている。これは、図 1 (C) に示すように、アイソクロナス gap の時間を、非同期転送のサブアクション gap の時間よりも短くすることで実現される。

【0039】1. 3 ツリー識別

ツリー識別はバスリセットの後に行われる。ツリー識別によりノード間の親子関係やルートノードが決定される。

【0040】まず、リーフノード（1つのノードにしか接続されていないノード）が、隣接するノードに、ペアレントノティファイを送る。例えば図 2 (A) のようにノード A、B、C、D、E が接続されている場合には、ノード A からノード B に、ノード D 及び E からノード C に、ペアレントノティファイ（PN）が送られる。

【0041】ペアレントノティファイを受け取ったノードは、送り元のノードを自身の子と認知する。そして、チャイルドノティファイをそのノードに送る。例えば図 2 (A) では、ノード B からノード A に、ノード C からノード D 及び E にチャイルドノティファイ（CN）が送られる。これによってノード B、A 間、ノード C、D 間、ノード C、E 間の親子関係が決定する。

【0042】ノード B、C の親子関係は、どちらが先にペアレントノティファイを送ったかで決定される。例えば図 2 (B) のように、ノード C の方が先にペアレントノティファイを送ると、ノード B が親になりノード C が子になる。

【0043】ポートの接続先の全てのノードが自身の子となるノードがルートになる。図 2 (B) では、ノード B がルートになる。なお、IEEE 1394 では、全てのノードがルートになる可能性がある。

【0044】1. 4 自己識別

ツリー識別の後、自己識別が行われる。自己識別においては、接続トポロジにおいてルートノードから遠いノードから順にセルフ ID packets が転送される。

【0045】より具体的には、例えば図 3 において、まず、ルートノード B のポート 1（番号の小さいポート）に接続されるノード A が、セルフ ID packets（自己識別 packets）を全てのノードにブロードキャストする。

【0046】次に、ルートノード B のポート 2（番号が大きいポート）に接続されるノード C が選択され、このノード C のポート 1（番号の小さいポート）に接続されるノード D がセルフ ID packets をブロードキャストする。次に、ノード C のポート 2（番号の大きいポート）に接続されるノード E がセルフ ID packets をブロードキャストし、その後、ノード C がブロードキャストする。最後に、ルートであるノード B がセルフ ID packets をブロードキャストし、自己識別が完了する。

【0047】セルフ ID packets には各ノードの ID が含まれる。ブロードキャストを行う時点で他のノードが

ら受け取ったセルフIDパケットの個数が、この各ノードのIDとなる。例えば図3では、ノードAがブロードキャストを行う時点では、どのノードもセルフIDパケットを発していないため、ノードAのIDは0になる。ノードAは、このID=0をセルフIDパケットに含ませてブロードキャストする。また、ノードDがブロードキャストを行う時点では、ノードAのみがセルフIDパケットを発している。このため、ノードDのIDは1になる。同様に、ノードE、C、BのIDは、各々、2、3、4になる。

【0048】図4(A)にセルフIDパケットのフォーマットを示す。同図に示すようにセルフIDパケットには各ノードの基本情報が含まれる。具体的には、各ノードのID(PHY_ID)、リンク層がアクティブか否か(L)、ギャップカウント(gap_cnt)、転送速度(sp)、アイソクロナスリソースマネージャになれる能力を有するか否か(C)、電力状態(pwr)、ポートの状態(p0、p1、p2)などに関する情報が含まれる。

【0049】なお、図4(B)に、ノードのポート数が4個以上の場合に使用されるセルフIDパケット#1、#2、#3のフォーマットを示す。ポート数が4~11個の場合にはセルフIDパケット#0(図4(A))及び#1が、12~19個の場合にはセルフIDパケット#0、#1及び#2が、20~27個の場合にはセルフIDパケット#0、#1、#2及び#3が使用されることになる。

【0050】また、図4(C)、(D)に、セルフIDパケットと同様に、物理層のパケット(PHYパケット)であるリンクオンパケット、PHY構成パケットのフォーマットを示す。

【0051】1.5 アイソクロナスリソースマネージャ
アイソクロナスリソースマネージャ(IRM)は以下の管理機能を有する。

【0052】第1に、アイソクロナス転送に必要な種々のリソースを提供する。例えば、チャネル番号レジスタや帯域幅レジスタを提供する。第2に、バスマネージャのIDを示すレジスタを提供する。第3に、バスマネージャがない場合に、簡易的なバスマネージャとなる役割を有する。

【0053】IRMになれる能力を有し(アイソクロナスリソースを管理する能力を有し)、且つ、動作状態になっている(リンク層がアクティブになっている)ノードの中で(IRMになれる資格を有するノードの中で)、ルートに最も近い(IDが最も大きい)ノードがIRMになる。より、具体的には、図4(A)のセルフIDパケットにおいて、IRMになれる能力を有するか否かを示すC(CONTENDER)ビットと、リンク層がアクティブか否かを示すL(LINK_ACTIVE

E)ビットが共に1になっているノードの中で、ルートに一番近いノード(PHY_IDが一番大きいノード)がIRMになる。例えば、ルートノードのセルフIDパケットのCビットとLビットが1の場合には、ルートノードがIRMになる。

【0054】1.6 サイクルマスタ、バスマネージャ
サイクルマスタは、図1(C)に示すサイクルスタートパケットを送信する役割を有し、ルートノードがサイクルマスタになる。

10 【0055】バスマネージャは、トポロジーマップ(各ノードの接続状態)の作成、スピードマップの作成、バスの電力管理、サイクルマスタの決定、ギャップカウンの最適化などの仕事を行う。

【0056】1.7 プロトコル構成

図5を用いて、IEEE1394のプロトコル構成(層構造)について説明する。

【0057】IEEE1394のプロトコルは、物理層、リンク層、トランザクション層により構成される。また、シリリアルバスマネージメントは、物理層、リンク層、トランザクション層をモニターしたり制御したりするものであり、ノードの制御やバスのリソース管理のための種々の機能を提供する。

【0058】物理層は、リンク層により使用されるロジカルシンボルを電気信号に変換したり、バスの調停を行ったり、バスの物理的インターフェースを定義する。

【0059】リンク層は、アドレッシング、データチェック、データフレーミング、サイクル制御などを提供する。

30 【0060】トランザクション層は、リード、ライト、ロックなどのトランザクションを行うためのプロトコルを定義する。

【0061】物理層及びリンク層は、通常、データ転送制御装置(インターフェースチップ)などのハードウェアにより実現される。また、トランザクション層は、CPU上で動作するファームウェアや、ハードウェアにより実現される。

【0062】2. 全体構成

次に、本実施形態の全体構成について図6を用いて説明する。

40 【0063】図6において、PHYインターフェース10は、物理層のプロトコルを実現するPHYチップとのインターフェースを行う回路である。

【0064】リンクコア20(リンク手段)は、リンク層のプロトコルやトランザクション層のプロトコルの一部を実現する回路であり、ノード間でのパケット転送のための各種サービスを提供する。レジスタ22は、これらのプロトコルを実現したりリンクコア20を制御するためのレジスタである。

50 【0065】FIFO(ATF)30、FIFO(ITF)32、FIFO(RF)34は、各々、非同期送信

用、アイソクロナス送信用、受信用のFIFOであり、例えばレジスタや半導体メモリなどのハードウェアにより構成される。本実施形態では、これらのFIFO 30、32、34の段数は非常に少ない。例えば1つのFIFOの段数は、好ましくは3段以下であり、更に好ましくは2段以下となる。

【0066】DMAC40、42、44は、各々、ATF、ITF、RF用のDMAコントローラである。これらのDMAC40、42、44を用いることで、CPU66に介入されることなく、RAM80とリンクコア20との間でのデータ転送が可能になる。なお、レジスタ46は、DMAC40、42、44などを制御するレジスタである。

【0067】ポートインターフェース50は、アプリケーション層のデバイス（例えばプリンタの印字処理を行うデバイス）とのインターフェースを行う回路である。本実施形態では、このポートインターフェース50を用いて、例えば8ビットのデータ転送が可能になっている。

【0068】FIFO(PF)52は、アプリケーション層のデバイスとの間でのデータ転送のためのFIFOであり、DMAC54は、PF用のDMAコントローラである。レジスタ56は、ポートインターフェース50やDMAC54を制御するレジスタである。

【0069】CPUインターフェース60は、データ転送制御装置をコントロールするCPU66とのインターフェースを行う回路である。CPUインターフェース60は、アドレスデコーダ62、データ同期化回路63、割り込みコントローラ64を含む。クロック制御回路68は、本実施形態で使用されるクロックを制御するものであり、PHYチップから送られてくるSCLKや、マスタークロックであるHCLKが入力される。

【0070】バッファマネージャ70は、RAM80とのインターフェースを管理する回路である。バッファマネージャ70は、バッファマネージャの制御のためのレジスタ72、RAM80へのバス接続を調停する調停回路74、各種の制御信号を生成するシーケンサ76を含む。

【0071】RAM80は、ランダムアクセス可能なパケット記憶手段として機能するものであり、その機能は例えばSRAM、DRAMなどにより実現される。そして、本実施形態では、図7に示すように、このRAM80がヘッダ領域（広義には制御情報領域）とデータ領域に分離されている。そして、パケットのヘッダ（広義には制御情報）は図7のヘッダ領域に格納され、パケットのデータはデータ領域に格納される。

【0072】なおRAM80は、本実施形態のデータ転送制御装置に内蔵させることが特に望ましいが、その一部又は全部を外付けにすることも可能である。

【0073】バス90（或いはバス92、94）は、ア

プリケーションに接続されるものである（第1のバス）。またバス96（或いはバス98）はデータ転送制御装置をコントロールするためのものであり、データ転送制御装置をコントロールするデバイス（例えばCPU）に電氣的に接続される（第2のバス）。またバス100（或いはバス102、104、105、106、107、108、109）は、物理層のデバイス（例えばPHYチップ）に電氣的に接続されるものである（第3のバス）。また、バス110は、ランダムアクセス可能な記憶手段であるRAMに電氣的に接続されるものである（第4のバス）。

【0074】バッファマネージャ70の調停回路74は、DMAC40、DMAC42、DMAC44、CPUインターフェース60、DMAC54からのバスアクセス要求の調停を行う。そして、この調停結果に基づいて、各々、バス105、107、109、98、94のいずれかと、RAM80のバス110との間にデータの経路が確立される（第1、第2、第3のバスのいずれかと第4のバスとの間にデータ経路が確立される）。

【0075】本実施形態の1つの特徴は、ランダムアクセスが可能でありパケットを格納するRAM80を設けると共に、互いに分離されるバス90、96、100と、これらのバスをRAM80のバス110に接続するための調停回路74とを設けた点にある。

【0076】例えば図8に、本実施形態と構成の異なるデータ転送制御装置の例を示す。このデータ転送制御装置では、リンクコア902は、PHYインターフェース900、バス922を介してPHYチップと接続される。また、リンクコア902は、FIFO904、906、908、CPUインターフェース910、バス920を介してCPU912に接続される。そして、CPU912は、バス924を介して、CPUにローカルなメモリであるRAM914に接続される。

【0077】なお、FIFO904、906、908は、図6のFIFO30、32、34と異なり、非常に段数の多いものとなる（例えば1つのFIFOが16段程度）。

【0078】図8の構成のデータ転送制御装置を用いた場合のデータ転送の手法について図9を用いて説明する。PHYチップ930を介して他のノードから送られてきた受信パケットは、バス922、データ転送制御装置932、バス920を介してCPU912が受け取る。そして、CPU912は、受け取った受信パケットをバス924を介してRAM914に書き込む。そして、CPU912は、受信パケットをアプリケーション層が使用できるように加工し、バス926を介してアプリケーション層のデバイス934に転送する。

【0079】一方、アプリケーション層のデバイス934からのデータを転送する場合には、CPU912は、このデータをRAM914に書き込む。そして、RAM

914のデータにヘッダを付加することでIEEE1394に準拠したパケットを生成する。そして生成されたパケットは、データ転送制御装置932、PHYチップ930などを介して他のノードに送信される。

【0080】しかしながら、このようなデータ転送手法によると、CPU912の処理負荷が非常に重くなる。従って、ノード間を接続するシリアルバスの転送速度が高速になっても、CPU912の処理のオーバーヘッドなどに起因して、システム全体の実転送速度は低くなり、結局、高速なデータ転送を実現できない。

【0081】このような問題を解決する1つの手法として、図10に示すように、データ転送制御装置932とRAM914との間でのデータ転送や、RAM914とアプリケーション層のデバイス934との間でのデータ転送を、ハードウェアDMAにより実現する手法も考えられる。

【0082】しかしながら、この手法では、CPUバス928が、データ転送制御装置932、RAM914間でのデータ転送、RAM914、CPU912間でのデータ転送、RAM914、アプリケーション層デバイス934間でのデータ転送に使用されることになる。従って、システム全体のデータ転送の高速化を図ろうとすると、CPUバス928としてPCIバスのような高速なバスを使用しなければならなくなり、これは、データ転送制御装置を使用する電子機器の高コスト化を招く。

【0083】これに対して、本実施形態では図11に示すように、データ転送制御装置120、アプリケーション層デバイス124間のバス90と、CPUバス96と、データ転送制御装置120、RAM80間のバス110とが互いに分離されている。従って、CPUバス96をデータ転送の制御のみに使用できるようになる。また、バス90を占有して、データ転送制御装置120、アプリケーション層デバイス124間でデータ転送を行うことができるようになる。例えば、データ転送制御装置120が組み込まれる電子機器がプリンタである場合には、バス90を占有して印字データを転送できるようになる。この結果、CPU66の処理負荷を軽減でき、システム全体の実転送速度を高めることができる。またCPU66として安価なものを採用できると共に、CPUバス96として高速なバスを使用する必要性がなくなる。このため、電子機器の低コスト化、小規模化を図れるようになる。

【0084】3. 送信パケットフォーマット

3.1 本実施形態の特徴

本実施形態では、図7に示すように、RAM80の記憶領域を、パケットのヘッダ（広義には制御情報）が格納されるヘッダ領域（広義には制御情報領域）とパケットのデータが格納されるデータ領域に分離している。

【0085】例えば図8の比較例では、CPU912は、送信する順序でFIFO904、906に送信パケ

ットを入力しなければならない。例えば、パケット1（ヘッダ1、データ1）、パケット2（ヘッダ2、データ2）、パケット3（ヘッダ3、データ3）を送信する場合には、ヘッダ1、データ1、ヘッダ2、データ2、ヘッダ3、データ3というような順序でFIFO904、906に送信パケットを入力しなければならない。従って、CPU912による並べ替え処理が必要になり、CPU912の処理負荷が非常に重くなる。これは結局、システム全体の実転送速度の低下という事態を招く。

【0086】これに対して、図6の本実施形態では、RAM80の記憶領域がヘッダ領域とデータ領域に分離される。より具体的には、図12に示すように、ヘッダ領域に格納されたヘッダとデータ領域に格納されたデータとがハードウェアにより結合され、各ノードに転送すべき送信パケットが組み立てられる。従って、CPU66の処理負荷が図8に比べて非常に軽くなり、システム全体の実転送速度を向上できる。また、CPU66として安価なものを採用できると共にCPU66に接続されるバスも低速なもので済むため、データ転送制御装置や電子機器の小規模化、低コスト化を図れるようになる。

【0087】また、本実施形態によれば、ヘッダはヘッダ領域にまとめられて格納され、データもデータ領域にまとめられて格納される。従って、ヘッダやデータの読み出し処理や書き込み処理の簡易化を図ることができ、処理のオーバーヘッドを軽減化できる。例えば、図11の手法でデータ転送を行う場合には、CPU66は、CPUバス96を介してヘッダ領域にのみアクセスしヘッダを読み出したり書き込んだりするだけでデータ転送を制御できるようになる。また、アプリケーション層のデバイス124は、データ領域のデータを連続してバス90を介して読み出したり、データを連続してデータ領域に書き込んだりすることが可能になる。

【0088】さて本実施形態におけるパケットのヘッダとデータの結合処理は、より具体的には例えば以下のようにして実現する。

【0089】即ち図13に示すように、本実施形態では、DMAC40（広義には読み出し手段）内のパケット結合回路280が読み出しアドレスRADRを指定する。そして、パケットのヘッダ（広義には制御情報）を、ランダムアクセス可能なRAM80のヘッダ領域（広義には制御情報領域）から読み出し、このヘッダと対をなすデータを、RAM80のデータ領域から読み出す。このようにして、ヘッダとデータが結合され、送信パケットが組み立てられる。そして、この送信パケットは、FIFO30、パケット転送のための各種サービスを提供するリンクコア20、及びPHYチップを介して各ノードに転送される。

【0090】ここで、ヘッダの作成及びヘッダ領域への書き込みは、CPU66のヘッダ作成部300が行う。

また、データ領域へのデータの取り込みは、データ取り込み指示部 302 からの指示にしたがい、例えばアプリケーション層のデバイスが行う。また、パケットの送信開始命令は、送信開始部 304 が発行する。なお、図 13 のヘッダ作成部 300、データ取り込み指示部 302、送信開始部 304、繰り返し回数設定部 306 の機能は、CPU 66 のハードウェアと、ファームウェアとにより実現される。

【0091】なお、図 14 に示すように、ヘッダ領域の各ヘッダには、データ領域からのデータの読み出しアドレスを指すデータポインタが、ヘッダ作成部 300 により付加されている。図 13 に示すように、パケット結合回路 280 は、ヘッダ領域から RDATA として読み出されたヘッダからこのデータポインタを取得し、取得したデータポインタを用いて、データ領域からデータを読み出す。このようにすることで、パケット結合回路 280 の読み出し処理を簡素化できる。

【0092】図 8 の構成では、送信する順序、即ちヘッダ、データの順序で FIFO に送信パケットを入力しなければならない。従って、図 15 (A) に示すように、まずファームウェアがヘッダの作成及び書き込みを行い、次にアプリケーション層のデバイスからのデータを取り込み、最後に送信開始を命令するというようなパターンでしか、パケットを送信できない。このため、処理の高速化を実現できない。

【0093】一方、本実施形態では、図 15 (B) に示すように、アプリケーション層のデバイスからのデータをデータ領域に取り込んでいる間に、ファームウェア（ヘッダ作成部 300）は、ヘッダの作成処理及びヘッダ領域への書き込み処理を行うことができる。そして、ファームウェア（送信開始部 304）は、データ取り込み処理とヘッダの書き込み処理の両方が終了したことを条件に、パケットの送信開始命令を発行する。このように本実施形態によれば、ファームウェアの処理負担を格段に軽減できると共に、処理の高速化を図れるようになる。

【0094】なお、図 16 (A) に比較例におけるファームウェアの処理のフローチャート例を示し、図 16 (B) に本実施形態におけるファームウェアの処理のフローチャート例を示す。

【0095】図 16 (A) では、ファームウェアは、ヘッダの作成及び書き込みを行った後に（ステップ S1）、データ取り込みを指示し（ステップ S2）、最後に送信開始命令を発行する（ステップ S3）。

【0096】一方、図 16 (B) では、ファームウェアは、まずデータの取り込み指示だけを行い（ステップ S11）、その後、ヘッダの作成及び書き込みを行う（ステップ S12）。そして、データの取り込みが終了したか否かを判断し（ステップ S13）、終了した場合には送信開始命令を発行する（ステップ S14）。このよう

にすることで、データの取り込み処理とヘッダの作成及び書き込み処理とを並列に行うことができるようになる。

【0097】なお、本実施形態では図 17 に示すように、RAM 80 のヘッダ領域を、受信用、送信用の領域に分離することが望ましい。また、データ領域を、受信用、送信用の領域に分離したり、アイソクロナス転送用、非同期転送用の領域に分離するようにしてもよい。また、ヘッダ領域とデータ領域の他に、これらの領域と分離された CPU 66 のワーク領域を設けることが望ましい。

【0098】更に、RAM 80 の記憶領域が複数の領域に分離される場合には、各領域のサイズを可変に制御することが望ましい。より具体的には図 17 に示すように、各領域の境界のアドレスを指すポインタ P1～P6 を可変に制御する。このようにすれば、アプリケーションに応じた最適な領域分割を実現できる。この場合、RAM 80 の各領域のサイズは、電源投入後もダイナミックに可変制御することが望ましい。このようにすれば、受信処理の際には受信用の領域を広くし、送信処理の際には送信用の領域を広くすることが可能になり、限られたリソースを有効利用できるようになる。

【0099】さて、図 18 (A) では、RAM に設けられる送信パケット領域が 1 つのチャンネルしか有していない。従って、図 18 (A) では、まず、この 1 つのチャンネルにパケット 1 を書き込み（ヘッダの作成及び書き込み、データの取り込み）、書き込みが完了するとパケット 1 の送信開始を命令する。次に、同じチャンネルにパケット 2 を書き込み、書き込みが完了するとパケット 2 の送信開始を命令することになる。

【0100】これに対して、図 18 (B) では、RAM の送信パケット領域が複数のチャンネルに分離されている。従って、図 18 (B) では、まずチャンネル CH1 にパケット 1 を書き込み、書き込みが完了するとパケット 1 の送信開始を命令する。そして、CH1 からパケット 1 を送信している間に、チャンネル CH2 にパケット 2 を書き込み、書き込みが完了するとパケット 2 の送信開始を命令する。同様に、CH2 からパケット 2 を送信している間に、CH3 にパケット 3 を書き込み、CH3 からパケット 3 を送信している間に、CH4 にパケット 4 を書き込む。

【0101】図 18 (A) では、パケットの書き込み処理と送信処理とがシーケンシャルに行われるため、処理時間に無駄が生じる。これに対して、図 18 (B) では、1 のチャンネルからのパケットの送信処理（読み出し処理）と、他のチャンネルへのパケットの書き込み処理とが並列に行われる。従って、処理時間の無駄を省くことができ、処理の高速化を図れる。

【0102】次に、複数チャンネルを用いたパケット送信の詳細について、図 19、図 20 (A)、(B)、図 2

1 (A)、(B)を用いて説明する。

【0103】図19に示すSUB1、SUB2、SUB3、SUB4は、各々、図20(A)、(B)、図21(A)、(B)に示すサブルーチン1、2、3、4を表すものである。そして、これらのSUB1～SUB4は全て、1つのメインルーチンから分岐する。

【0104】まず、図19のE1に示すようにチャンネルCH1においてSUB1の処理が行われる。SUB1の処理では、図20(A)に示すように、データ取り込み中フラグがセットされているか否かを判断し(ステップT11)、セットされていない場合にはデータ領域に空きがあるか否かを調べる(ステップT12)。そして空きがある場合には、データの取り込みを指示する(ステップT13)。これにより、アプリケーション層のデバイス等からRAMにデータが取り込まれる。そして、データ取り込みの指示の後、データ取り込み中フラグをセットする(ステップT14)。

【0105】SUB2の処理では、図20(B)に示すように、まずデータ取り込み中フラグがセットされているか否かを判断する(ステップT21)。そして、図19のE2では、E1のSUB1のステップT14(図20(A))においてデータ取り込み中フラグがセットされている。従って、次のステップに移行し、データの取り込みが終了しているか否かを判断する(ステップT22)。そして、図19のE2では、データの取り込みが終了しているため、次のステップに移行し、チャンネルCH1の送信可能フラグをセットする(ステップT23)。この送信可能フラグは、前述のデータ取り込み中フラグや後述する送信中フラグと異なり、チャンネル毎に設けられる。送信可能フラグをセットした後、データ取り込み中フラグをクリアする(ステップT24)。

【0106】SUB3の処理では、図21(A)に示すように、まず送信中フラグがセットされているか否かを判断し(ステップT31)、セットされていない場合には送信可能フラグがセットされているか否かを判断する(ステップT32)。そして、図19のE3では、E2のSUB2のステップT23(図20(B))において、送信可能フラグがセットされている。従って、次のステップに移行し、送信の開始を命令する(ステップT33)。そして、送信開始命令の後に、送信中フラグをセットする(ステップT34)。

【0107】SUB4の処理では、図21(B)に示すように、まず送信中フラグがセットされているか否かを判断する(ステップT41)。そして、図19のE4では、E3のSUB3のステップT34(図21(A))において送信中フラグがセットされている。従って、次のステップに移行し、送信が終了しているか否かを判断する(ステップT42)。そして、図19のE4では、送信が既に終了しているため、次のステップに移行し、送信可能フラグ及び送信中フラグをクリアする(ステッ

プT43、T44)。

【0108】図19のE2のSUB2では、図20

(B)のステップT22、T23に示すように、データの取り込みが終了したこと条件に、そのチャンネルの送信可能フラグがセットされる。そして、E3のSUB3では、図21(A)のステップT32、T33に示すように、送信可能フラグがセットされたことを条件に、そのチャンネルからの送信が開始される。即ち、このような送信可能フラグをチャンネル毎に設けることで、そのチャンネルのデータ取り込みが終了したことを条件にそのチャンネルの送信を開始できるようになる。

【0109】また図19のE2のSUB2は、図20

(B)のステップT22、T24に示すように、データの取り込みが終了したことを条件にデータ取り込み中フラグをクリアする。そして、このようにデータ取り込み中フラグがクリアされると(セットされていないと)、E5のSUB1が、データの取り込みを指示できるようになる(図20(A)のステップT11、T13)。即ち、このようなデータ取り込み中フラグを使用することで、1のチャンネルのデータ取り込み処理が行われている間に他のチャンネルのデータ取り込み処理が行われてしまうという事態を回避できる。

【0110】また図19のE6のSUB4は、図21

(B)のステップT42、T44に示すように、送信が終了したことを条件に送信中フラグをクリアする。そして、このように送信中フラグがクリアされると(セットされていないと)、E7のSUB3が、送信開始を命令できるようになる(図21(A)のステップT31、T33)。即ち、このような送信中フラグを使用することで、1のチャンネルの送信処理が行われている間に他のチャンネルの送信処理が行われてしまうという事態を回避できる。

【0111】以上のようにすることで、メインルーチンとサブルーチンSUB1～SUB4とからなる1つのプログラム(ファームウェア)を用いて、複数チャンネルからのパケットの送信処理をマルチタスクのように実行できるようになる。

【0112】なお、複数チャンネルを用いたパケット転送においては、各パケットが、パケット同士を連結するための連結ポインタを有することが望ましい。例えば図22(A)では、連結ポインタ1、2、3により、各々、パケット1及び3、2及び4、3及び2が連結される。そして、パケット1の送信開始が命令されると、連結ポインタ1によりパケット3が読み出され、図22(B)に示すようにパケット3が送信される。そして、パケット3が送信されると、連結ポインタ3によりパケット2が読み出され、図22(C)に示すようにパケット2が送信される。そして、パケット2が送信されると、連結ポインタ2によりパケット4が読み出され、図22(D)に示すようにパケット4が送信される。

【0113】このようにすれば、ファームウェアが1のチャンネルの packets の送信開始を命令するだけで、他のチャンネルの packets が連続して順次読み出され送信されるようになる。従って、ファームウェアは、他のチャンネルの packets については送信開始を命令する必要がなくなる。この結果、ファームウェアの処理負担を格段に軽減できる。

【0114】さて、IEEE1394では、図1(A)で説明したように、ソースノードが packets を転送すると、デスティネーションノードはソースノードにACKコードを返す。この場合、このACKコードの内容を、ソースノードのファームウェアが確認できるように何らかの記憶手段に格納しておくことが望まれる。そして、図23(A)では、packet 1 の送信に対して返ってきた4ビットのACKコードを、4ビットのレジスタ940に格納している。そして、packet 2 の送信に対してACKコードが返ってきた場合には、このpacket 2 についてのACKコードをレジスタ940に上書きする。即ち、レジスタ940のACKコードは、常に最新のものに更新される。

【0115】しかしながら、図23(A)の手法では、複数の packets を連続して送信した場合に、どの packets のACKコードがレジスタ940に格納されているのかがわからなくなる。レジスタ940には、1つのACKコードしか格納できないからである。

【0116】この問題を回避する1つの手法として、packet 1 の送信に対して返ってきたACKコードを、ファームウェアが、レジスタ940から読み出して確認した後に packet 2 の送信開始を命令するという手法が考えられる。しかしながら、この手法では、ファームウェアは、packet 1 のACKコードが返ってくるまで、packet 2 の送信開始命令の発行を待たなければならない。このため、処理の高速化が妨げられる。特に、図18(B)のように複数チャンネルを用いて packets を送信する場合に、複数チャンネルを有することの利点を生かすことができなくなる。

【0117】そこで図23(B)では、packet の転送先(デスティネーションノード)から返送されてくるACKコード(アクノリジメント情報)を、送信 packet 領域の複数のチャンネルの中の、その packet の送信元のチャンネルに書き戻すようにしている。即ち、図23(B)に示すように、チャンネルCH1の packet 1 の送信に対して返ってきたACKコードはCH1に書き戻し、チャンネルCH2の packet 2 の送信に対して返ってきたACKコードはCH2に戻す。

【0118】このようにすることで、各 packet と返送されてきたACKコードとが一对一に対応するようになり、どの packet に対してどのACKコードが返ってきたかを、ファームウェアは簡易且つ確実に確認できるようになる。

【0119】また、図23(C)に示すように、ファームウェアが、packet 1 の送信に対して返ってくるACKコードを確認せずに packet 2 の送信開始を命令し、その後に、返ってきたACKコードを確認するようにすることもできる。即ち、送信の準備(ヘッダの生成及び書き込み、データの取り込み)が整った段階ですぐに送信開始を命令し、空いている時間を見計らって、後でACKコードを確認するようにする。このようにすることで、更なる処理の高速化を図れるようになる。

【0120】なお、ハードウェアの簡素化、ファームウェアの処理負担の軽減化、処理の高速化の観点からは、図23(B)、(C)のように packet のACKコードを送信元のチャンネルに書き戻す手法を採用することが特に望ましい。しかしながら、本実施例は図23(B)、(C)の手法に限られるものではない。即ち、packet の転送先から返送されてくるACKコードを、返送されてくるACKコードを確認せずに連続転送できる packet 数以上の個数だけ記憶できる手段があればよい。例えば、ACKコードを確認せずに連続転送できる packet 数が4個である場合には、4個以上のACKコードを記憶できるレジスタ(16ビット以上のレジスタ)を用意する。そして、望ましくは、どのような順番でACKコードが返ってきたのかを確認できるようにしておく。このようにすれば、packet を連続転送した場合にも、packet とACKコードとの対応関係を維持しておくことが可能になる。

【0121】さて、図24(A)では、packet 1 (ヘッダ1、データ1)をRAMに書き込み、その送信の開始を命令し、packet 2 (ヘッダ2、データ2)を書き込み、その送信の開始を命令し、packet 3 (ヘッダ3、データ3)を書き込み、その送信の開始を命令するというようにして、packet 1、2、3を送信している。

【0122】しかしながら、packet 1、2、3が同種の packet であり、ヘッダ1、2、3が同様のものである場合(例えば大きなサイズのデータを複数のペイロードに分割して送信する場合)には、図24(A)の手法では、ファームウェアのヘッダ作成処理が無駄であると共に、RAMの記憶領域も無駄に使用されてしまう。また、packet 1、2、3を送信するために、ファームウェアは3回の送信開始命令を発行しなければならない。従って、図24(A)では、ファームウェアの処理負担が重くなると共に、限られたリソースを有効利用できない。

【0123】これに対して、図24(B)では、図13のヘッダ作成部300(ファームウェア)が基本となるヘッダを1つ作成する。この基本ヘッダは、図24

(B)に示すようにデータポインタ、RPN、t1などを含む。

【0124】ここでデータポインタは、データ領域から

読み出されるデータのアドレスを指すものである。また RPN は、図 13 の繰り返し回数設定部 306 により設定される送信繰り返し回数情報である。また t1 (トランザクションラベル) は、各ノードからのトランザクションを識別するための情報として IEEE1394 で規格化されているものである。応答側は、要求側からのパケットに含まれるトランザクションラベルと同一のものを応答側のパケットに含ませて、要求側に返す必要がある。従って、各トランザクション毎に t1 の書き換え処理が必要になる。

【0125】図 13 の送信開始部 304 が送信開始を命令すると、ヘッダ書き換え回路 310 が、図 24 (B) に示すよう基本ヘッダを順次書き換える。即ち、基本ヘッダの RPN をデクリメント又はインクリメント (広義には更新) しながら、基本ヘッダのデータポイントや t1 を書き換える。この書き換えは、RPN が例えば零 (所与の値) になるまで行われる。このように基本ヘッダを書き換えることで、連続転送されるパケットのデータと対をなすヘッダが、パケットを送信する毎に順次作成される。このようにすることで、図 24 (A) に比べて、ファームウェアの処理負担を格段に軽減できると共に、ヘッダが占有する記憶領域の大きさも小さくできる。

【0126】例えば、図 24 (A) の手法では、図 25 (A) に示すように、パケットを送信する毎に、ファームウェアは、ヘッダの作成及び書き込みと、送信開始の命令とを行わなければならない。これに対して、図 24 (B) の手法では、図 25 (B)、図 26 に示すように、ファームウェアは、初めに RPN の設定 (ステップ U1) と基本ヘッダの作成及び書き込み (ステップ U2) を行い、送信開始を命令すれば (ステップ U3)、その後は、自動的に基本ヘッダが書き換えられて、データの取り込み、送信が繰り返される。従って、図 25 (A) に比べてファームウェアの処理負担を軽減できる。

【0127】なお、送信繰り返し回数情報 RPN を、基本ヘッダに含ませないで、所与のレジスタなどに格納するようにしてもよい。

【0128】3. 2 構成

次に送信側の構成について説明する。図 27 に、FIFO 30、DMAC 40 の詳細な構成の一例を示す。

【0129】位相調整用のバッファとして機能する FIFO 30 は、FIFO 状態判断回路 31 を含む。FIFO 状態判断回路 31 は、FIFO が空になると、EMPTY をアクティブにし、FIFO がフルになると、FULL をアクティブにする。

【0130】DMAC 40 は、パケット結合回路 280、アクセス要求実行回路 290、アクセス要求発生回路 292、ACK 書き込み要求発生回路 294、ACK 書き込みデータ&アドレス発生回路 296 を含む。

【0131】パケット結合回路 280 は、ヘッダを RAM80 のヘッダ領域から読み出し、データをデータ領域から読み出し、これらのヘッダとデータによりフレームが構成される送信パケットを組み立てる回路である (図 12 参照)。パケット結合回路 280 は、ポインタ更新回路 284、アドレス発生回路 288 を含む。

【0132】ポインタ更新回路 284 は、RAM80 からヘッダやデータを読み出すためのヘッダポインタ (広義には制御情報ポインタ) やデータポインタを更新するための回路であり、データポインタ取得回路 285 を含む。データポインタ取得回路 285 は、RAM80 から読み出された RDATA からデータポインタを取得する回路であり、tcode 判別回路 286 を含む。tcode 判別回路 286 は、tcode (トランザクションコード。広義にはパケットフォーマット識別情報) を判別する回路であり、この tcode を判別することで、パケットのヘッダとデータのいずれが読み出されているかを判断できる。

【0133】アドレス発生回路 288 は、ポインタ更新回路 284 の出力などを受け、RAM80 の読み出しアドレス RADR を発生する回路である。

【0134】アクセス要求実行回路 290 は、FIFO 状態判断回路 31 からの EMPTY がアクティブになると、FIFOIN をアクティブにする。リンクコア 20 は、FIFOIN がアクティブでないことを条件に、TD (TxData) のストロブ信号である TDS をアクティブにする。

【0135】なお TFAIL は、送信における失敗を、リンクコア 20 がアクセス要求実行回路 290 に対して知らせるための信号である。

【0136】アクセス要求発生回路 292 は、バッファマネージャ 70 からの読み出しアクノリジメントである RACK や FIFO 状態判断回路 31 からの FULL を受け、読み出し要求である RREQ をバッファマネージャ 70 に出力する。

【0137】ACK 書き込み要求発生回路 294 は、リンクコア 20 からの TCMPI やバッファマネージャ 70 からの WACK を受け、WREQ をバッファマネージャ 70 に出力する。また、ACK 書き込みデータ&アドレス発生回路 296 は、リンクコア 20 からの TACK を受け、送信パケットのチャネルに書き戻す ACK のコードを WDATA として出力し、ACK を書き戻すアドレスを WADR として出力する。

【0138】3. 3 送信側の動作

次に、送信側の動作の詳細について図 28 のタイミング波形図などを用いて説明する。

【0139】まず、リンクコア 20 の動作について説明する。

【0140】送信開始を知らせる TSTART がアクティブになると、図 28 の B1 に示すように、リンクコア

20は、ストローブ信号であるTDSを用いてFIFO30からTDを取り込む。この場合、リンクコア20には、ヘッダ(H0~H3)、データ(D0~Dn)の順でTDSが取り込まれる。

【0141】なお、図29(A)に、シリアルバス上で転送される非同期パケットのフォーマット(IEEE1394規格)を示す。一方、図29(B)に、RAM80のヘッダ領域に格納される非同期送信パケットのヘッダ部分のフォーマットを示す。同図に示すようにヘッダの第4クワドレットはデータポインタになっている。

【0142】さて、図28のB2に示す位置では、リンクコア20は、TDSをアクティブにしない。従って、B3に示すように、ヘッダの第4クワドレットであるH4はリンクコア20に取り込まれない。図29(B)に示すように第4クワドレットのH4はデータポインタであり、リンクコア20は、このデータポインタを必要としないからである。そして、リンクコア20は、B3に示す期間において、ヘッダCRC(図29(A)参照)を生成しヘッダに付加する処理を行っている。

【0143】1つのパケットの送信処理が終了すると、リンクコア20は、B4に示すようにTCMPをアクティブにする。そして、送信のデスティネーションノードからPHYチップを介して返されてきたACKのコードを(図1(A)、図23(B)、(C)参照)、B5に示すようにTACKとしてDMAC40に出力する。このACKのコードは、ACK書き込み要求発生回路294、ACK書き込みデータ&アドレス発生回路296により、RAM80のヘッダ領域のヘッダに書き戻されることになる(図29(B)の第7クワドレット)。

【0144】次に、FIFO30の動作について説明する。

【0145】FIFO30は、バッファマネージャ70からのRDATAを受け、TDとしてリンクコア20に出力する。

【0146】FIFO30内のFIFO状態判断回路31は、内蔵するカウンタにより、FIFO30のデータ数(FIFOカウント)をカウントする。そして、FIFO30が空(データ数=0)になった場合には、図28のB6に示すようにEMPTYをアクティブにする。また、FIFO30がフル(データ数=2)になった場合には、B7に示すようにFULLをアクティブ(Hレベル)にする。FIFO30が空になったことは、EMPTY、FIFOINにより、DMAC40内のアクセス要求実行回路290やリンクコア20に伝えられる。また、FIFO30がフルになったことは、FULLにより、DMAC40内のアクセス要求発生回路292に伝えられる。

【0147】次に、DMAC40の動作について説明する。

【0148】アクセス要求発生回路292は、B8に示

すように、FULLが非アクティブ(Lレベル)であること(FIFO34がフルでないこと)を条件にRREQをアクティブにする。そして、バッファマネージャ70からRACKを受け取ると、RREQを非アクティブにする。

【0149】なお、本実施形態では、送信時のバス調停においては、DMAC40(或いはDMAC42)からのアクセス要求の優先順位を最も高くしている。従って、DMAC40からのRREQと、CPUインターフェース60やポート用のDMAC54からのアクセス要求(OtherRREQ)とが競合した場合には、RREQの方が優先する。一方、RREQよりも先に、CPUインターフェース60やポート用のDMAC54からのアクセス要求があった場合には、B9に示すように、DMAC40のアクセス要求は所与の期間だけ待たされる。従って、バッファマネージャ70からのRDATAと、リンクコア20へのTDとは同期しない。この理由により、本実施形態では、RDATAとTDの位相を調整するためのFIFO30を設けている。この場合、FIFO30は、位相調整のために必要な最低限の段数(好ましくは3段以下、更に好ましくは2段以下)を備えていればよい。

【0150】さて、送信が開始すると、ポインタ更新回路284が、図30(A)に示すようにヘッダポインタHPのインクリメント(広義には更新)を行う。そして、アドレス発生回路288は、図28のB10に示すように、インクリメントされるヘッダポインタに応じたRADRを発生する。このようにして、RDATAのヘッダ部分がRAM80から順次読み出される。

【0151】RDATAとしてH4が読み出されると、パケット結合回路280に含まれるデータポインタ取得回路285が、このH4を、データポインタDPとして取得する。より具体的には、RDATAとしてH0が読み出されると、データポインタ取得回路285内のtcod判別回路286が、H0に含まれるtcod(図29(B)参照)を判別する。そして、ヘッダの例えば第4クワドレットにデータポインタがあるとtcod(広義にはパケットフォーマット識別情報)に基づき判断された場合には、RDATAとしてH4が読み出された時に、データポインタ取得回路285がこのH4を取得する。即ち、図28のB11に示すように、RDATAのH4が、データポインタとして取得され、RADRとして出力される。

【0152】なお、本実施形態では、B3、B11に示すように、リンクコア20がヘッダCRCを生成している期間を利用して、RDATAからデータポインタであるH4を取得している。即ち、本実施形態では、ヘッダCRCの生成はリンクコア20が行い、DMAC40はこれに関与しない。一方、データポインタの取得はDMAC40が行い、リンクコア20はこれに関与しない。

本実施形態では、このことに着目し、図29(A)においてヘッダCRCが配置される第4クワドレットに、図29(B)に示すようにデータポインタを配置している。そして、ヘッダCRCが生成される期間を利用して、RDATAからデータポインタであるH4を取得するようにしている。このようにすることで、処理時間に無駄が生じるのを防止できるようになる。

【0153】データポインタが取得されると、ポインタ更新回路284が、図30(B)に示すように、取得されたデータポインタであるH4のインクリメントを行う。そして、アドレス発生回路288は、図28のB12に示すように、インクリメントされるデータポインタに応じたRADRを発生する。このようにして、RDATAのデータ部分がRAM80から順次読み出される。

【0154】1つのパケットの送信処理が終了し、B4に示すようにTCMPがアクティブになると、ACK書き込み要求発生回路294がB13に示すようにWREQをアクティブにする。そして、リンクコア20からACK書き込みデータ&アドレス発生回路296に対してTACKを用いて送られたACKのコードが、B14に示すように、WDATAとして出力される。また、この際に、ACKのコードの書き込みアドレスであるHP+7が、WADRとして出力される。このようにすることで、図23(B)、(C)で説明したように、パケットの送信元のチャンネルに対して、デスティネーションノードからのACKコードを書き戻すことが可能になる。

【0155】なお、WADRがHP+7になるのは、図29(B)に示すようにACKのコードはヘッダの第7クワドレットに書き戻されるからである。

【0156】以上のようにして、ヘッダ領域からのヘッダとデータ領域からのデータを結合して、送信パケットを組み立てることが可能になる。

【0157】特に本実施形態では、ヘッダとデータの結合はDMAC40が行い、リンク20はこれに関与する必要がない。従って、リンクコア20の回路構成や処理を簡素化できるようになる。

【0158】また、本実施形態では、データポインタ取得回路285が、RDATAからデータポインタ(H4)を取得し、この取得されたデータポインタに基づきRADRが発生し、データが読み出される。このようにすることで、ヘッダと、そのヘッダに対応するデータとを、適切に結合できるようになる。また、ヘッダとデータの結合処理に必要な回路構成を簡素化できるようになる。

【0159】なお、ヘッダ領域とデータ領域の境界などの、RAM80を分離する領域の境界(図17のP1~P6)の設定は、CPUインターフェース60を介してCPU66(ファームウェア等)が、図6のレジスタ46に含まれるポインタ設定レジスタに対して、境界のアドレスを指すポインタを設定することで実現される。

【0160】4. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器の例について説明する。

【0161】例えば図31(A)に電子機器の1つであるプリンタの内部ブロック図を示し、図32(A)にその外観図を示す。CPU(マイクロコンピュータ)510はシステム全体の制御などを行う。操作部511はプリンタをユーザが操作するためのものである。ROM516には、制御プログラム、フォントなどが格納され、RAM518はCPU510のワーク領域として機能する。表示パネル519はプリンタの動作状態をユーザに知らせるためのものである。

【0162】PHYチップ502、データ転送制御装置500を介して、パーソナルコンピュータなどの他のノードから送られてきた印字データは、バス504を介して印字処理部512に直接送られる。そして、印字データは、印字処理部512にて所与の処理が施され、プリントヘッドなどからなる印字部(データを出力するための装置)514により紙に印字されて出力される。

【0163】図31(B)に電子機器の1つであるスキャナの内部ブロック図を示し、図32(B)にその外観図を示す。CPU520はシステム全体の制御などを行う。操作部521はスキャナをユーザが操作するためのものである。ROM526には制御プログラムなどが格納され、RAM528はCPU520のワーク領域として機能する。

【0164】光源、光電変換器などからなる画像読み取り部(データを取り込むための装置)522により原稿の画像が読み取られ、読み取られた画像のデータは画像処理部524により処理される。そして、処理後の画像データがバス505を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、この画像データにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0165】図31(C)に電子機器の1つであるCD-Rドライブの内部ブロック図を示し、図32(C)にその外観図を示す。CPU530はシステム全体の制御などを行う。操作部531はCD-Rをユーザが操作するためのものである。ROM536には制御プログラムなどが格納され、RAM538はCPU530のワーク領域として機能する。

【0166】レーザ、モータ、光学系などからなる読み取り&書き込み部(データを取り込むための装置又はデータを記憶するための装置)533によりCD-R532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、バス506を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、このデータにヘッダなどを

付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0167】一方、PHYチップ502、データ転送制御装置500を介して、他のノードから送られてきたデータは、バス506を介して信号処理部534に直接送られる。そして、信号処理部534によりこのデータに所与の信号処理が施され、読み取り&書き込み部533によりCD-R532に記憶される。

【0168】なお、図31(A)、(B)、(C)において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

【0169】本実施形態のデータ転送制御装置を電子機器に用いることで、高速なデータ転送が可能になる。従って、ユーザがパーソナルコンピュータなどによりプリントアウトの指示を行った場合に、少ないタイムラグで印字が完了するようになる。また、スキャナへの画像取り込みの指示の後に、少ないタイムラグで読み取り画像をユーザは見るができるようになる。また、CD-Rからのデータの読み取りや、CD-Rへのデータの書き込みを高速に行うことができるようになる。更に、例えば1つのホストシステムに複数の電子機器を接続して利用したり、複数のホストシステムに複数の電子機器を接続して利用したりすることも容易になる。

【0170】また本実施形態のデータ転送制御装置を電子機器に用いることで、CPU上で動作するファームウェアの処理負荷が軽減され、安価なCPUや低速のバスを用いることが可能になる。更に、データ転送制御装置の低コスト化、小規模化を図れるため、電子機器の低コスト化、小規模化も図れるようになる。

【0171】なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例えば、種々の光ディスクドライブ(CDROM、DVD)、光磁気ディスクドライブ(MO)、ハードディスクドライブ、TV、VTR、ビデオカメラ、オーディオ機器、電話機、プロジェクタ、パーソナルコンピュータ、電子手帳、ワードプロセッサなど種々のものを考えることができる。

【0172】なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0173】例えば、本発明のデータ転送制御装置の構成は、図6に示す構成が特に望ましいが、これに限定されるものではない。例えば、送信パケット領域を複数チャンネルに分離する発明においては、記憶手段を制御情報領域とヘッダ領域とに分離しないようにしてもよい。また、アクノリッジメント情報を確認せずに連続転送できるパケット数以上の個数だけ、アクノリッジメント情報を記憶しておく発明においては、図6のみならず、図8などの種々の構成を採用することができる。

【0174】また、本発明は、IEEE1394規格でのデータ転送に適用されることが特に望ましいが、これに限定されるものではない。例えばIEEE1394と同様の思想に基づく規格やIEEE1394を発展させた規格におけるデータ転送にも本発明は適用できる。

【0175】

【図面の簡単な説明】

【図1】図1(A)、(B)、(C)は、非同期転送とアイソクロナス転送について説明するための図である。

10 【図2】図2(A)、(B)は、ツリー識別について説明するための図である。

【図3】自己識別について説明するための図である。

【図4】図4(A)、(B)、(C)、(D)は、セルブIDパケットなどの物理層のパケットのフォーマットを示す図である。

【図5】IEEE1394のプロトコル構成について示す図である。

【図6】本実施形態のデータ転送制御装置の構成例を示す図である。

20 【図7】ヘッダ(制御情報)領域とデータ領域の分離について説明するための図である。

【図8】本実施形態の比較例の構成例について示す図である。

【図9】図8の構成によるデータ転送の手法について説明するための図である。

【図10】データ転送の手法の他の例について説明するための図である。

【図11】本実施形態のデータ転送の手法について説明するための図である。

30 【図12】ヘッダ領域に格納されるヘッダとデータ領域に格納されるデータを結合し、送信パケットを組み立てる手法について説明するための図である。

【図13】RAMから読み出されたヘッダからデータポインタを取得し、取得したデータポインタを用いて、パケットのヘッダとデータを結合する手法について説明するための図である。

【図14】ヘッダ領域に格納するヘッダにデータポインタを含ませる手法について説明するための図である。

40 【図15】図15(A)、(B)は、データの取り込み処理の間に、ヘッダの作成及び書き込み処理を行う手法について説明するための図である。

【図16】図16(A)、(B)は、図15(A)、(B)の場合のファームウェアの処理を説明するためのフローチャートである。

【図17】RAMの各領域のサイズを可変に制御する手法について説明するための図である。

【図18】図18(A)、(B)は、送信パケット領域を複数チャンネルに分離する手法について説明するための図である。

50 【図19】送信パケット領域を複数チャンネルに分離した

場合のファームウェアの処理について説明するための図である。

【図 20】図 20 (A)、(B) は、送信パケット領域を複数チャネルに分離した場合のファームウェアの処理について説明するためのフローチャートである。

【図 21】図 21 (A)、(B) も、送信パケット領域を複数チャネルに分離した場合のファームウェアの処理について説明するためのフローチャートである。

【図 22】図 22 (A)、(B)、(C)、(D) は、連結ポインタを用いる手法について説明するための図である。

【図 23】図 23 (A)、(B)、(C) は、パケットの送信元のチャネルに ACK コードを書き戻す手法について説明するための図である。

【図 24】図 24 (A)、(B) は、基本ヘッダを書き換えてパケットを連続転送する手法について説明するための図である。

【図 25】図 25 (A)、(B) は、基本ヘッダを書き換えてパケットを連続転送する手法の利点について説明するための図である。

【図 26】基本ヘッダを書き換えてパケットを連続転送する場合のファームウェアの処理について説明するためのフローチャートである。

【図 27】送信側の構成の一例を示す図である。

【図 28】送信側の動作について説明するためのタイミング波形図である。

【図 29】図 29 (A) は、IEEE 1394 規格の非同期パケットのフォーマットであり、図 29 (B) は、RAM のヘッダ領域に格納される非同期送信パケットのヘッダ部分のフォーマットである。

【図 30】図 30 (A)、(B) は、ヘッダポインタとデータポインタの更新について説明するための図である。

【図 31】図 31 (A)、(B)、(C) は、種々の電子機器の内部ブロック図の例である。

【図 32】図 32 (A)、(B)、(C) は、種々の電子機器の外観図の例である。

【符号の説明】

10 PHY インターフェース

20 リンクコア

22 レジスタ

* 30 FIFO (ATF)

32 FIFO (ITF)

34 FIFO (RF)

40 DMAC (ATF 用)

42 DMAC (ITF 用)

44 DMAC (RF 用)

50 ポートインターフェース

52 FIFO (PF)

54 DMAC (PF 用)

10 56 レジスタ

60 CPU インターフェース

62 アドレスデコーダ

63 データ同期化回路

64 割り込みコントローラ

66 CPU

68 クロック制御回路

70 バッファマネージャ

72 レジスタ

74 調停回路

20 76 シーケンサ

80 RAM (パケット記憶手段)

90、92、94 バス (第 1 のバス)

96、98 バス (第 2 のバス)

100、102、104、105、

106、107、108、109 バス (第 3 のバス)

110 バス (第 4 のバス)

120 データ転送制御装置

122 PHY チップ

124 アプリケーション層のデバイス

30 280 パケット結合回路

284 ポインタ更新回路

285 データポインタ取得回路

286 t c o d e 判別回路

290 アクセス要求実行回路

292 アドレス要求発生回路

294 ACK 書き込み要求発生回路

296 ACK 書き込みデータ & アドレス発生回路

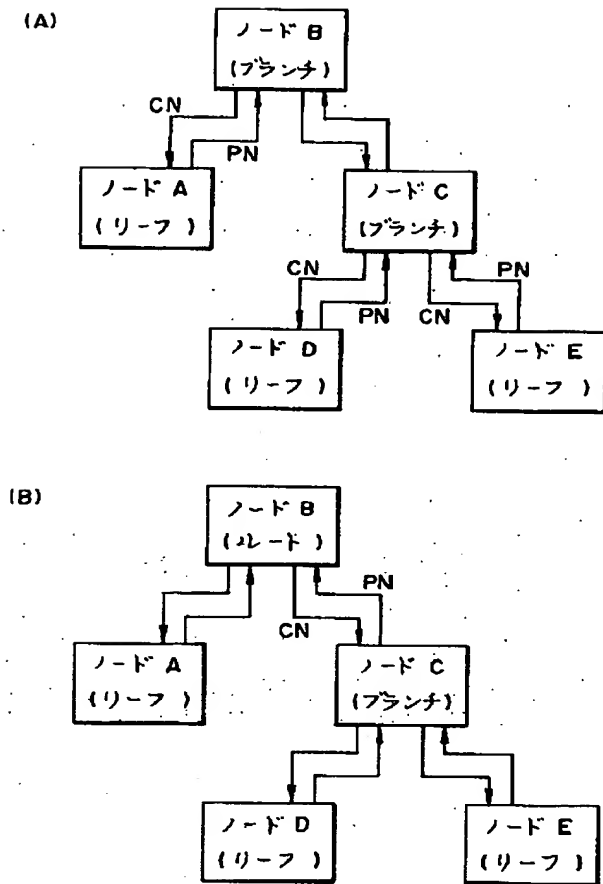
300 ヘッダ作成部

302 データ取り込み指示部

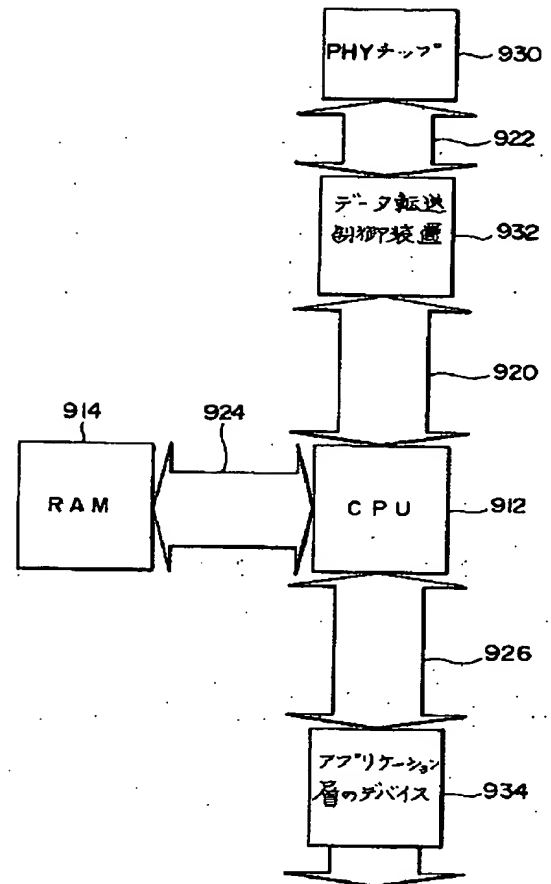
40 304 送信開始部

* 306 繰り返し回数設定部

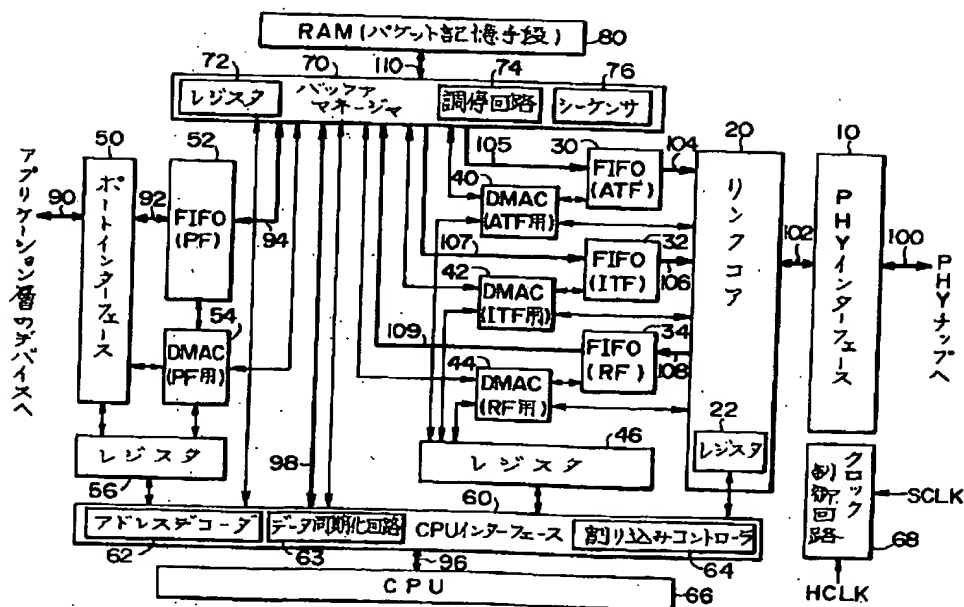
【図2】



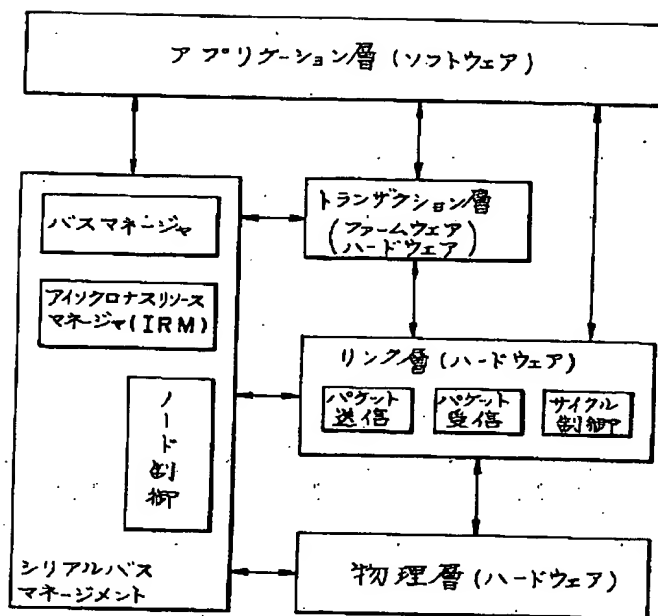
【図9】



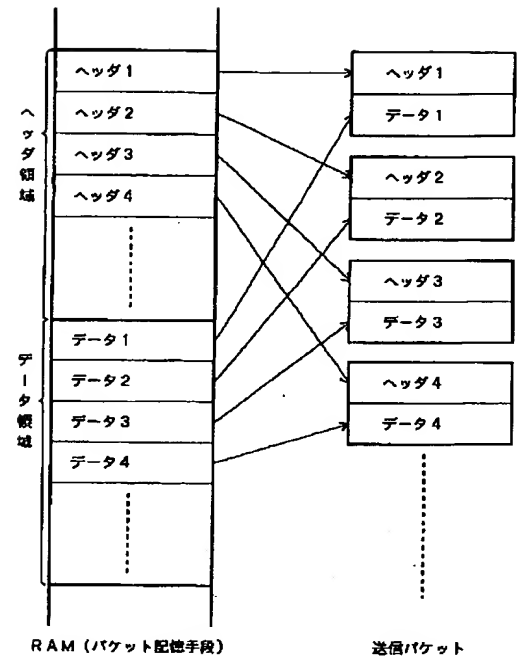
【図6】



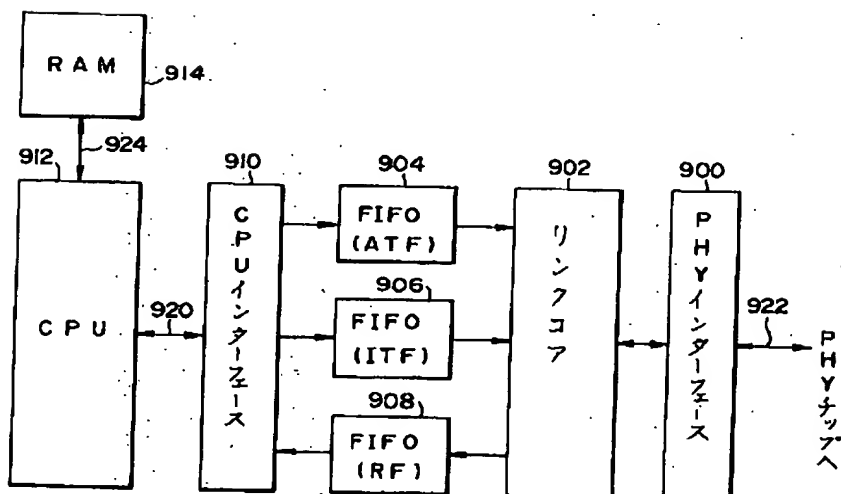
【図 5】



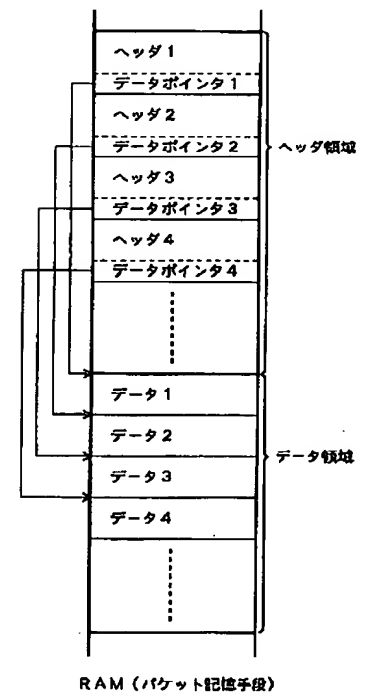
【図 12】



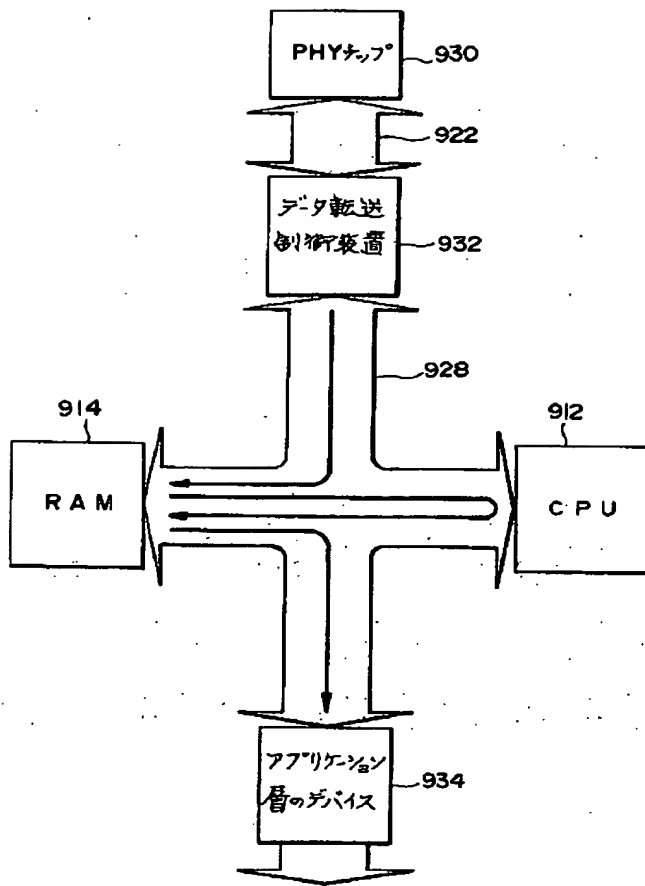
【図 8】



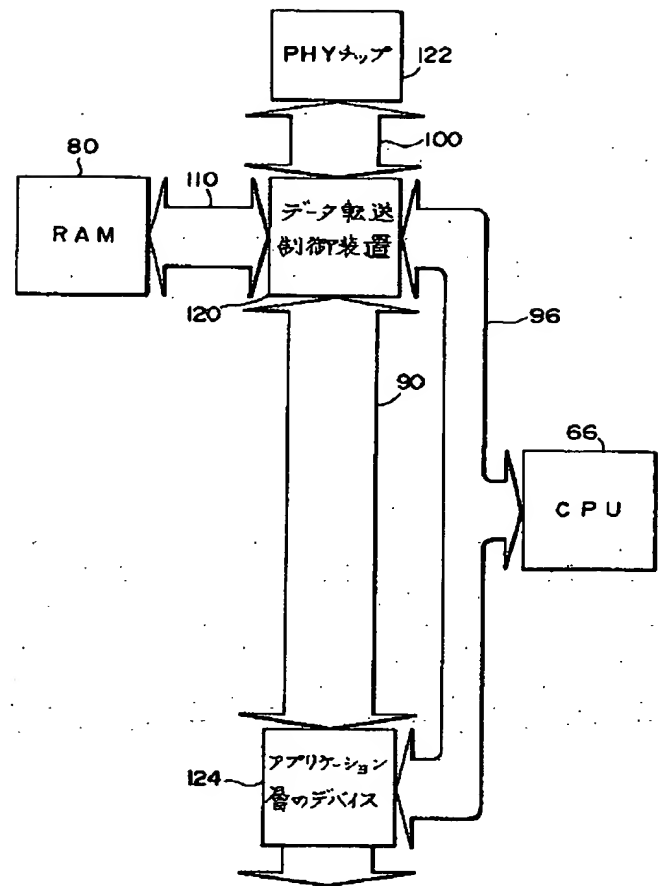
【図 14】



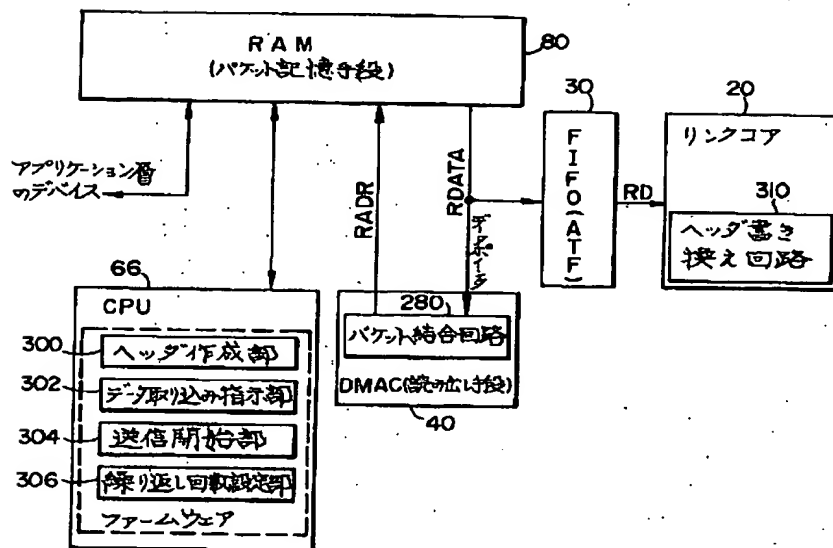
【図10】



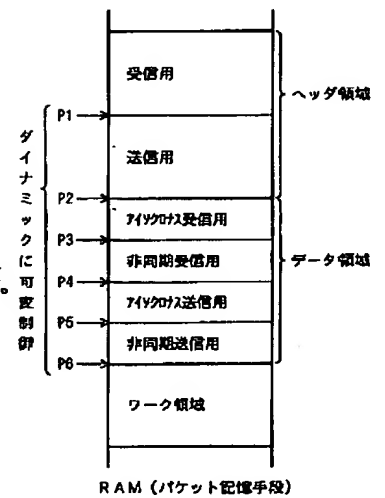
【図11】



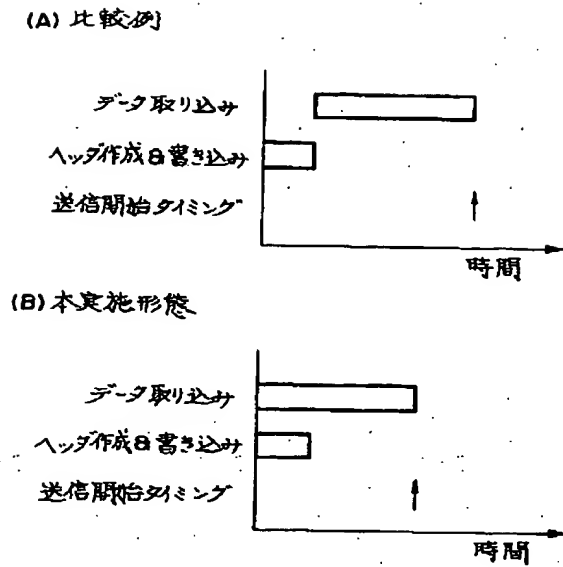
【図13】



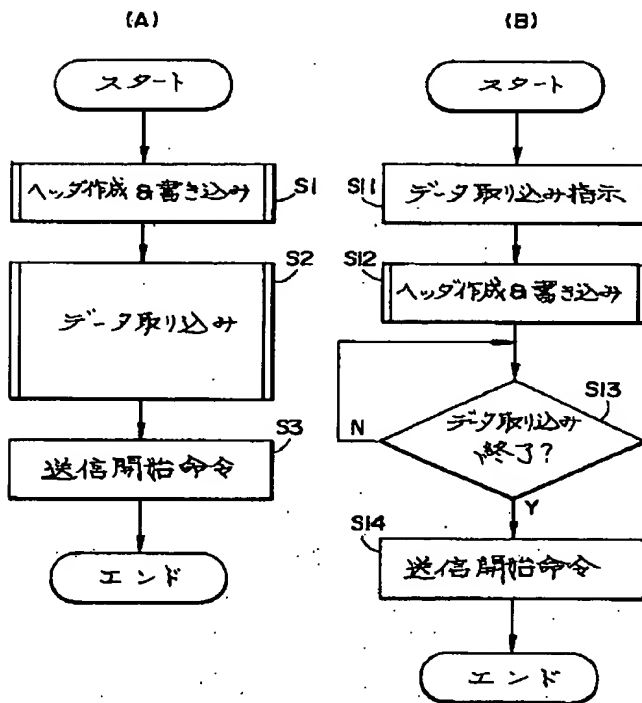
【図17】



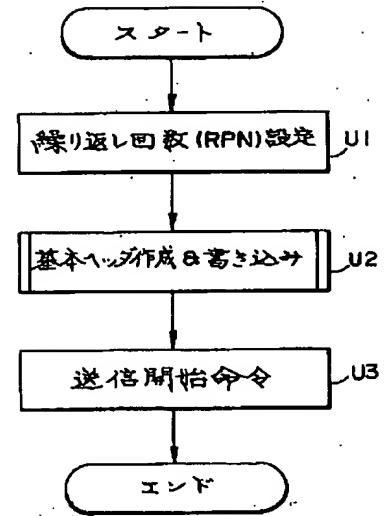
【図15】



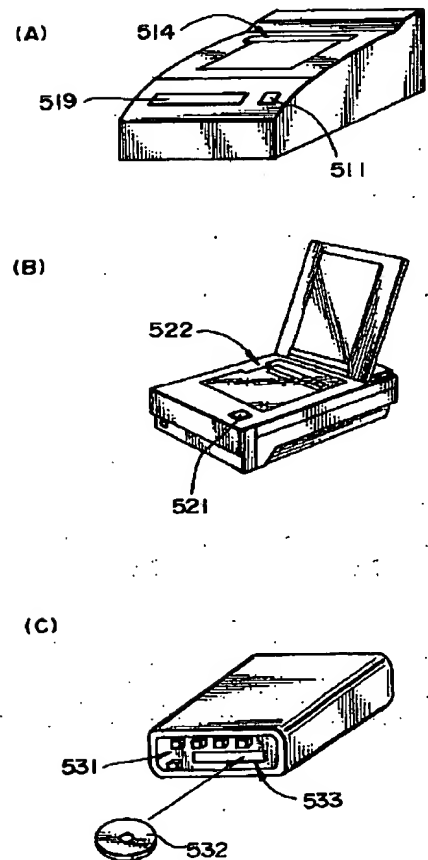
【図16】



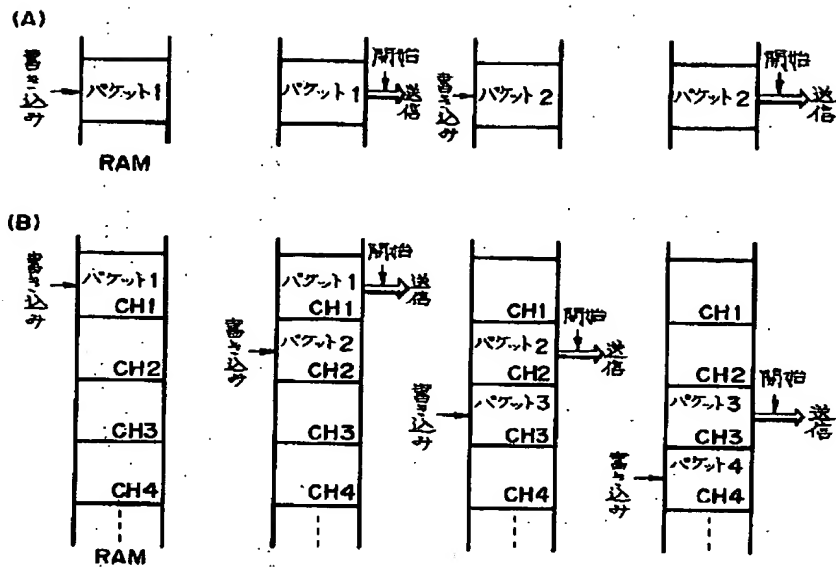
【図26】



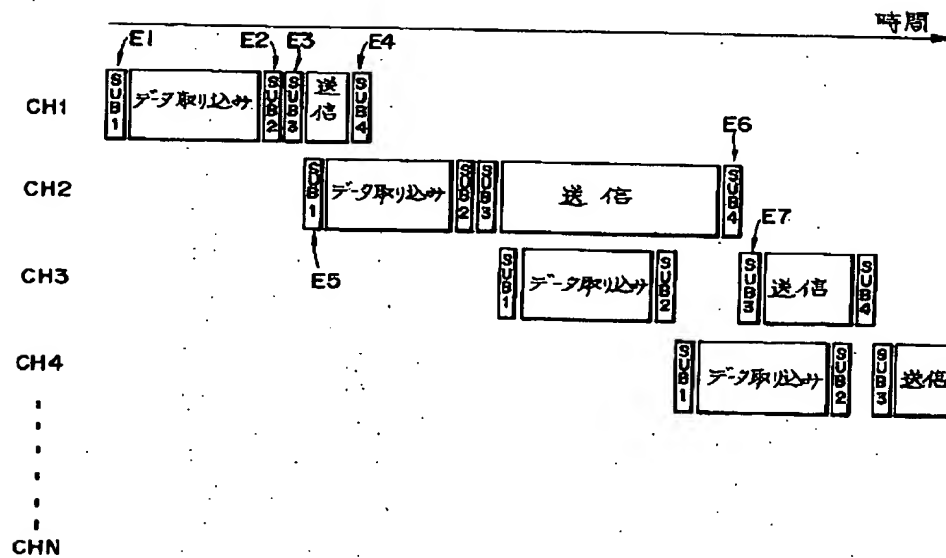
【図32】



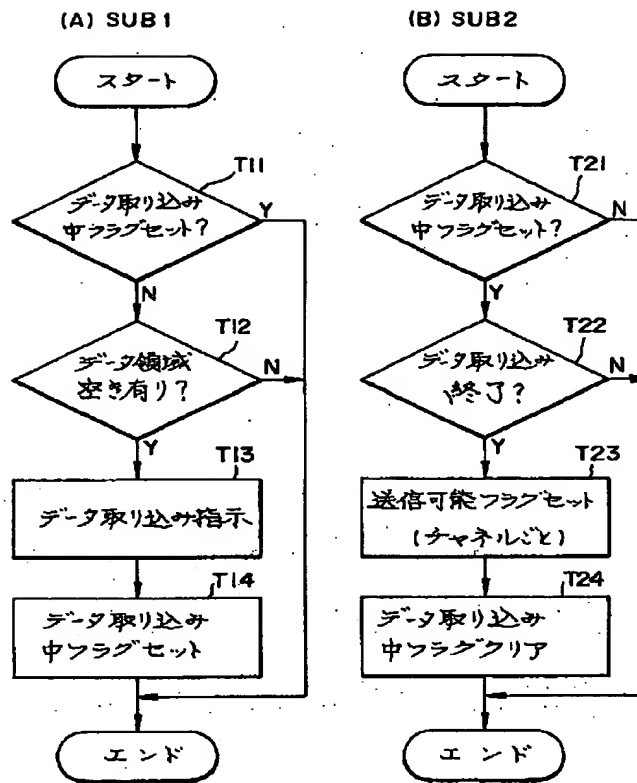
【図18】



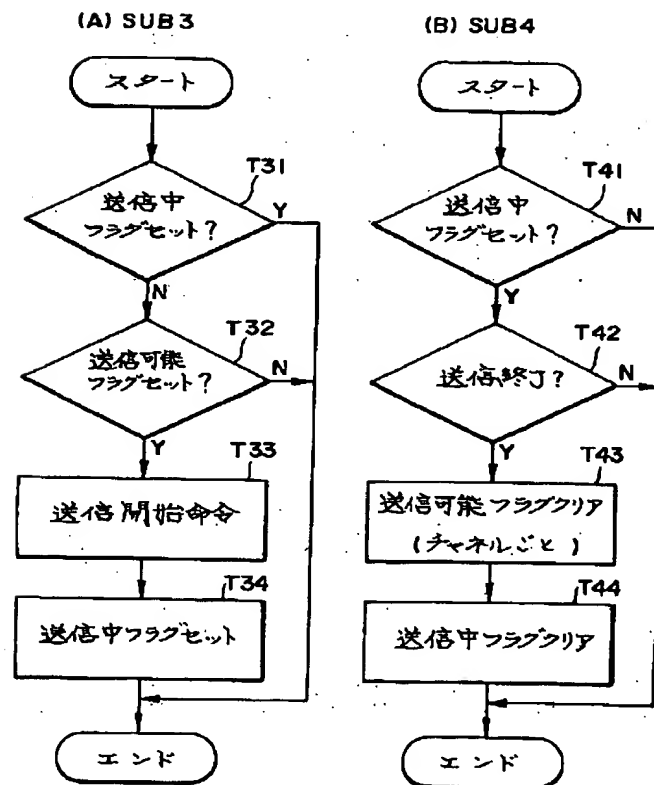
【図19】



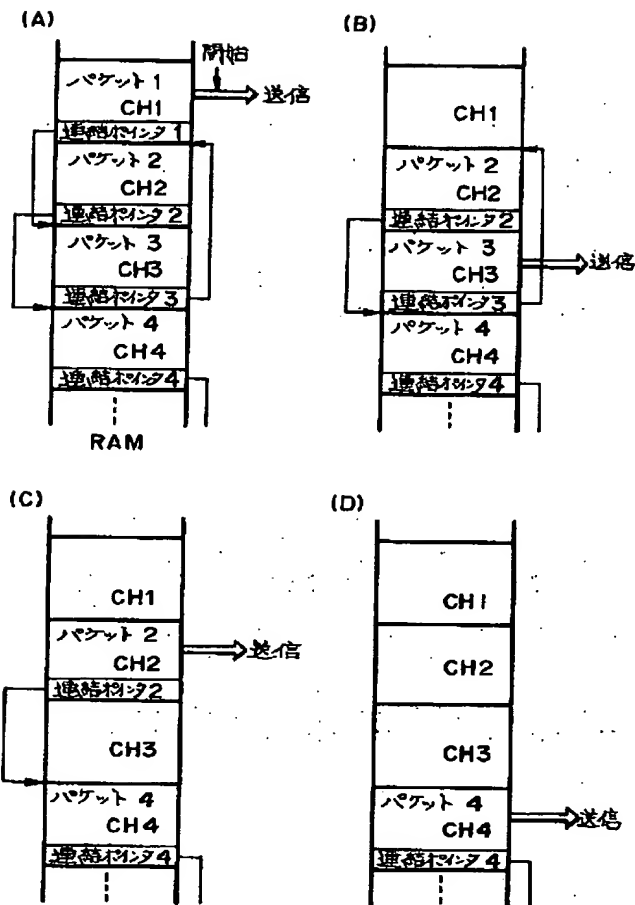
【図20】



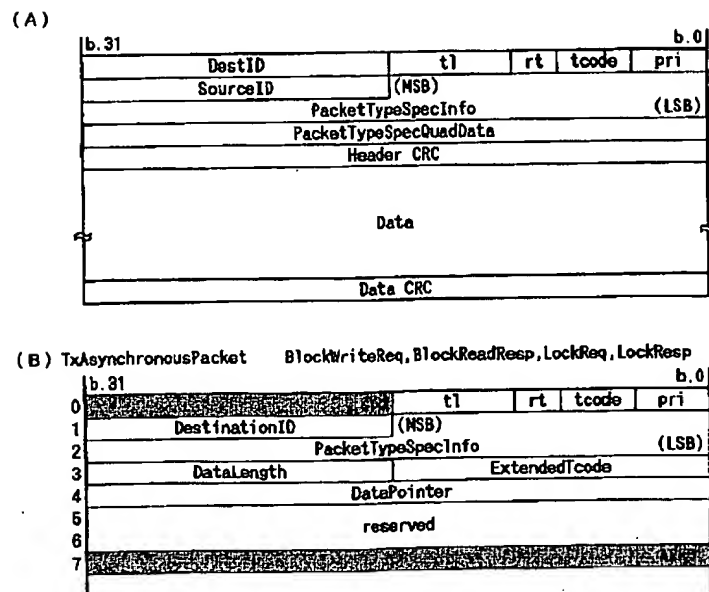
【図21】



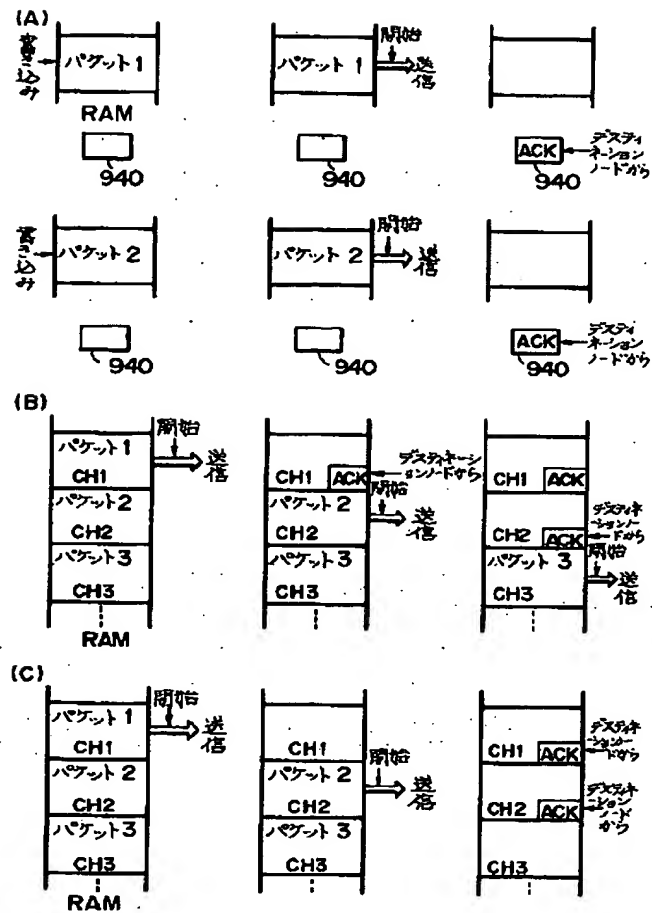
【図 22】



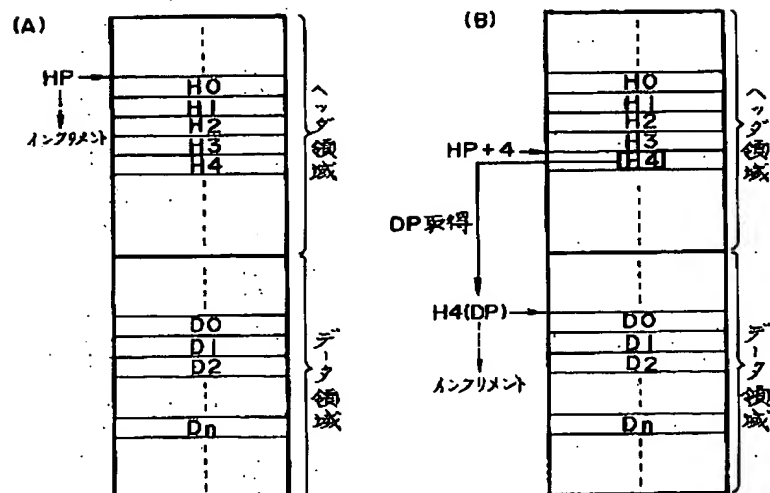
【図 29】



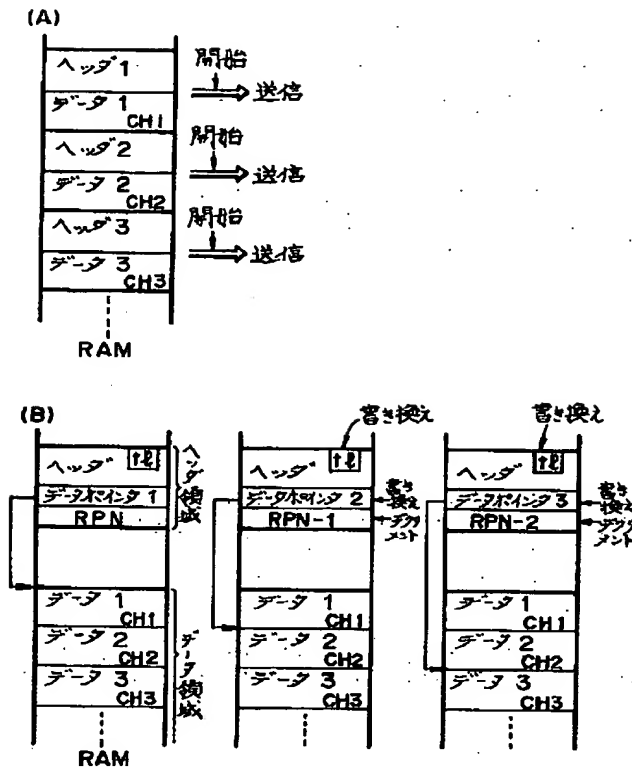
【図23】



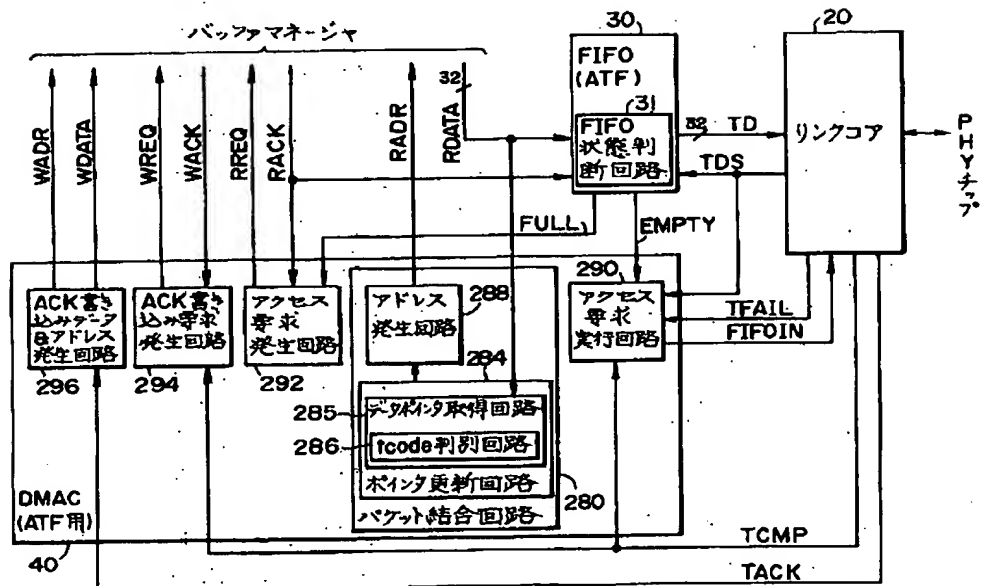
【図30】



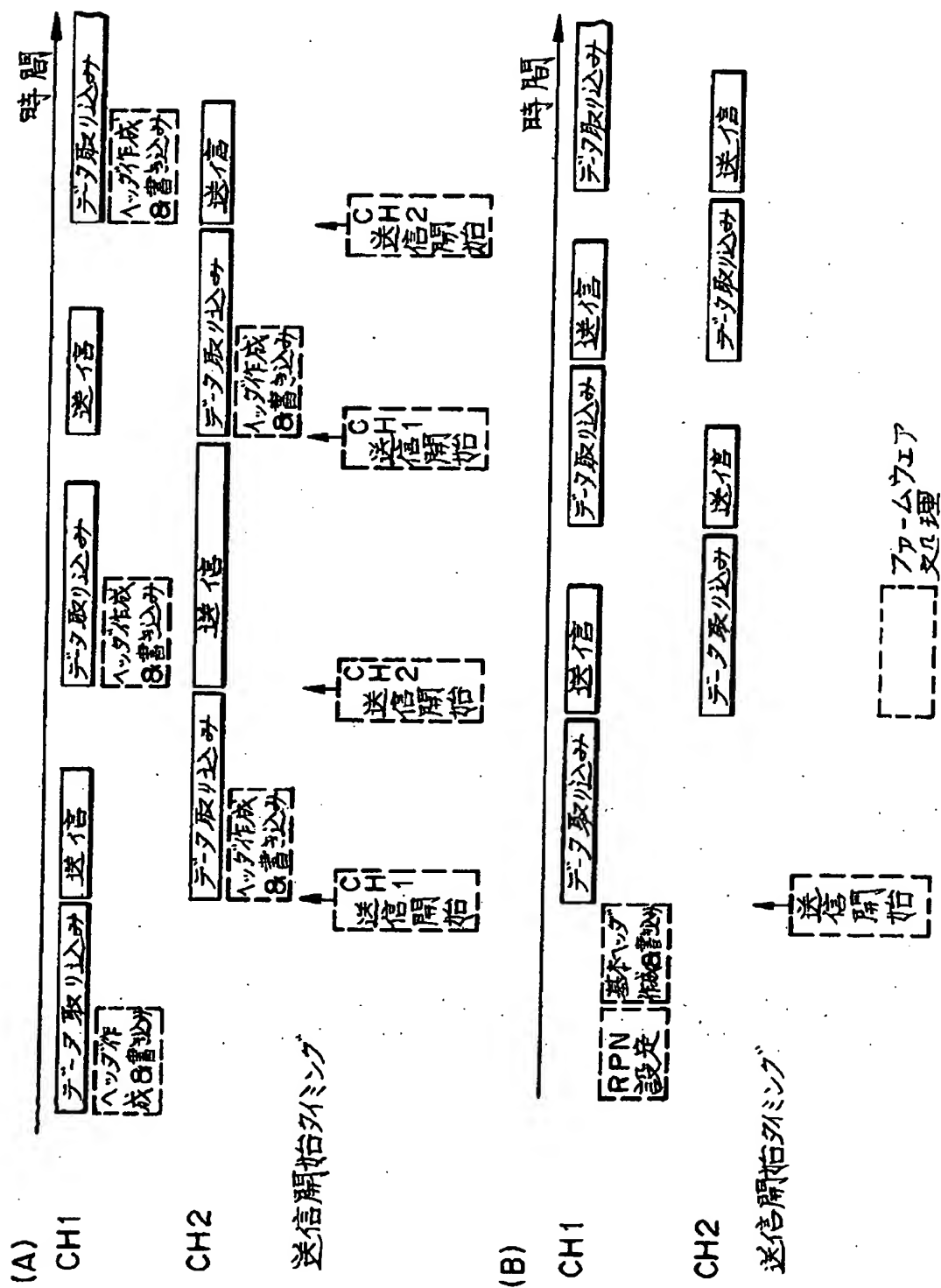
【図24】



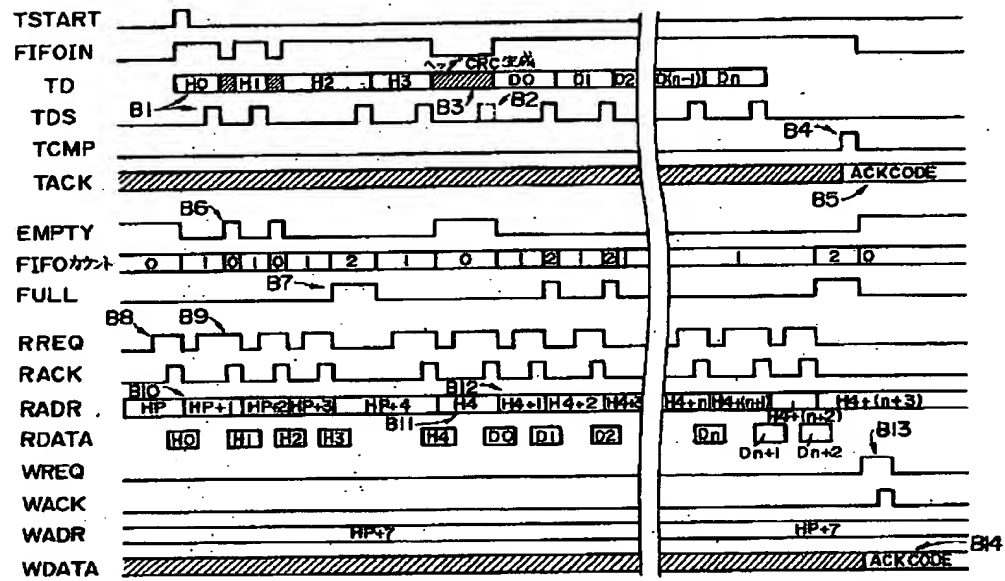
【図27】



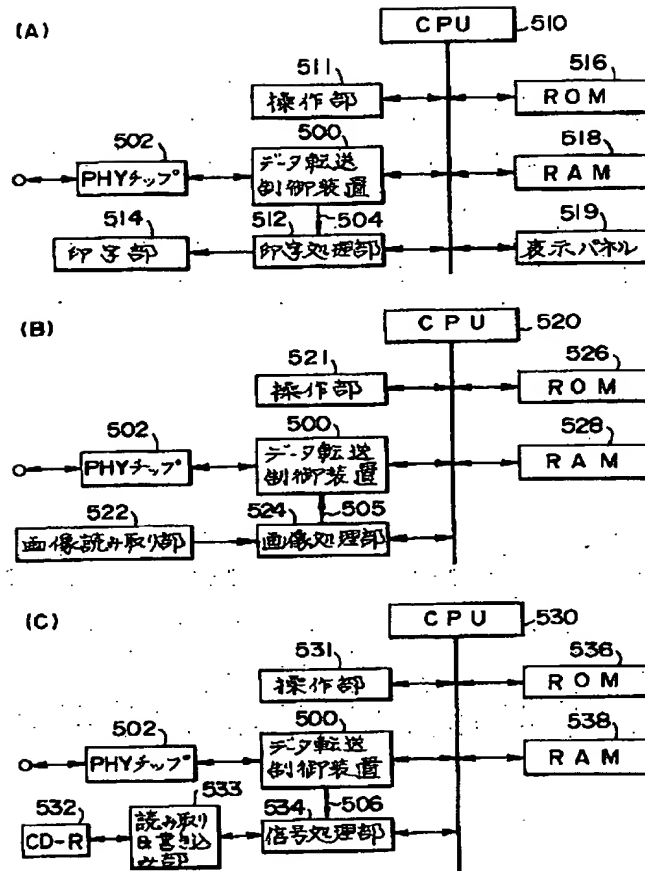
【图 2 5】



【図 28】



【図31】



フロントページの続き

(72)発明者 和田 文利
 長野県諏訪市大和3丁目3番5号 セイコ
 ーエプソン株式会社内

Fターム(参考) 5K032 AA02 AA03 CC03 CD01 DB19
 DB24